`bls`: An R package for analysing population genetic models with balancing selection, selective sweeps, and demographic changes

Kai Zeng

Department of Animal and Plant Sciences, University of Sheffield
(k.zeng@sheffield.ac.uk)

Contents

1 Introduction

This package contains functions and R6 classes for analysing several population genetic models using the phase-type approach detailed in Zeng *et al.* (2020). Briefly, for each model, functions for calculating the expected total branch length and the site frequency spectrum (SFS) are provided. As detailed in Zeng *et al.* (2020), the total branch length can be used to calculate the expected nucleotide diversity, the expected number of segregating sites, and σ^2, a measure of linkage disequilibrium.

2 Obtaining and installing the package

The package can be downloaded from `http://zeng-lab.group.shef.ac.uk/`. It is distributed as a `tar.gz` file that contains the source code and help files. Launch `R`. Issue the following command

```
install.packages(path_to_tarball, repos = NULL, type = "source")
```

Note that the process takes a little while (about 25 minutes on my computer), with most of the time spent on the step "byte-compile and prepare package for lazy loading". Once done, issue `library(bls)` to load the package. Issue `help(package = "bls")` to access documentation of the package.

3 Population genetic models

3.1 A model with long-term balancing selection and changes in population size

Consider a diploid, randomly mating population. Looking back in time, its evolutionary history consists of H non-overlapping epochs, with the most recent epoch being epoch 1. The effective population size is $N_{e,h}$ in epoch h ($h \in \{1, 2, ..., H\}$). The duration of epoch h is t_h generations. Epoch H, the most ancestral epoch, has an infinite time span, over which the population is at statistical equilibrium. We assume that an autosomal locus has two alleles A_1 and A_2. The *backward* mutation rate between A_i and A_j is v_{ij} per generation (see Zeng *et al.* (2020) for definition). This locus is under strong balancing selection, such that the equilibrium frequencies of A_1 and A_2 are $\hat{p}_1$ and $\hat{p}_2$, respectively. This set-up can accommodate any model of long-term balancing selection (with or without reversible mutation between A_1 and A_2), as long as it produces stable allele frequencies. Furthermore, we assume that selection is sufficiently strong, and the changes in population size are sufficiently small, that the frequencies of the two alleles remain at $\hat{p}_1$ and $\hat{p}_2$ in the more recent epochs. As shown in the paper, this assumption is robust even when selection is relatively weak.

A random sample of n alleles have been taken from a linked neutral locus. The recombination frequency between this locus and the selected locus is denoted by r. We are interested in obtaining diversity patterns at the neutral locus.

Let $N_{e,r}$ be a reference diploid population size. Scale time in units of $2N_{e,r}$ generations. The dynamics in epoch h are determined by the following parameters: (1) $\hat{p}_1$; (2) $M_{ij} = \mu_{ij} + \rho\hat{p}_j$, where $\mu_{ij} = 2N_{e,r}v_{ij}$ and $\rho = 2N_{e,r}r$; (3) $g_h = N_{e,r}/N_{e,h}$; (4) $d_h = t_h/(2N_{e,r})$.

Note that $\hat{p}_1$ and M_{ij} are shared across epochs. As argued in Zeng *et al.* (2020), in most cases it suffices to use a simplified model with $M_{ij} = \rho\hat{p}_j$.

3.1.1 An example equilibrium model of balancing selection

This model is composed of a single epoch of infinite length. The effective population size N_e is constant over time. Using this as the reference effective population size, the model has the following parameters: $\hat{p}_1$, M_{12}, and M_{21}.

3.1.1.1 Total branch length

```
# Create a new object. n1 is the number of alleles associated with A1 in the
   sample. See documentation for more details.
n <- 10
n1 <- 1
bls <- BranchLength2D$new(n, n1)

p1 <- 0.35
mu12 <- 0 # The simplified model
mu21 <- 0
rho <- 2
g <- 1 # This parameter is for accommodating changes in population size and is
    not relevant here.
d <- Inf # The equilibrium model is composed of a single epoch of infinite
   length.

# Obtain the total branch length
longTermBlsL(bls, p1, mu12, mu21, rho, g, d)

# Get result for a different sampling configuration
n1 <- 0
bls$setN1(n1)
longTermBlsL(bls, p1, mu12, mu21, rho, g, d)

# Get results for all possible sampling configurations, ordered as (0, n), (1,
    n - 1), ..., (n, 0).
n1 <- -1
bls$setN1(n1)
longTermBlsL(bls, p1, mu12, mu21, rho, g, d)
```

3.1.1.2 Site frequency spectrum

Consider a sample of n alleles at a linked neutral site, with n_1 alleles associated with A_1 and n_2 alleles associated with A_2. Let X_i be the total length of all branches in the genealogy that lead to i alleles in the sample ($0 < i < n$). Evidently $\sum_{i=1}^{n-1} X_i$ is the total branch length. If we define $\theta = 2N_{e,r}u$, where u is the neutral mutation rate per generation. Then under the infinite-sites model, θX_i is the expected number of polymorphic sites in the sample at which the derived allele appears i times. Note that the sample size is limited to $n \leq 12$ (see Section 4 for performance issues).

The model is parameterised slightly differently. Instead of having g_h and $\hat{p}_1$ as above, we define a vector $\boldsymbol{g}_h = (N_{e,r}/(N_{1,h}\hat{p}_1), N_{e,r}/(N_{2,h}\hat{p}_2))$. The following example uses the same selection model as the example presented in the previous section.

```
# Create a new object. n1 is the number of alleles associated with A1 in the
    sample.
n <- 10
n1 <- 1
bls <- Sfs2D$new(n, n1)

# Prepare the object
bls$reset()

p1 <- 0.35
rho <- 2
d <- Inf
M12 <- rho * (1 - p1) # The simplified model without reversible mutation
M21 <- rho * p1
g <- c(1 / p1, 1 / (1 - p1))
bls$addEpoch(M12, M21, g, d)

# Get the SFS (i.e., X_i)
bls$getSFS()

# If n1 is changed, we need to add the epoch again
n1 <- -1
bls$setN1(n1)
bls$addEpoch(M12, M21, g, d)

# When n1 = -1, the first n + 1 rows in the returned matrix are the SFSs for
    all the possible sampling configurations, ordered as (0, n), (1, n - 1),
    ..., (n, 0)
sfs <- bls$getSFS()
```

3.1.2 An example model of balancing selection with variable population size

Figure 1 shows the epochs and the population sizes. To scale time and the other parameters, we arbitrarily set $N_{e,r} = 1,000$. At the selected locus, the equilibrium frequency of A_1 is $\hat{p}_1 = 0.35$. There is no reversible mutation between A_1 and A_2 (i.e., the simplified model with $\mu_{ij} = 0$).

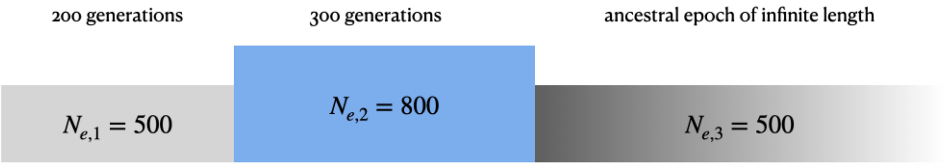

Figure 1: A model with variable migration rates and deme sizes.

3.1.2.1 Total branch length

```
# Create a new object. n1 is the number of alleles associated with A1 in the
    sample. See documentation for more details.
n <- 10
n1 <- 1
bls <- BranchLength2D$new(n, n1)

p1 <- 0.35
mu12 <- 0 # The simplified model
mu21 <- 0
rho <- 2
g <- numeric(3)
d <- numeric(3)

Ner <- 1000 # Reference population size

# Epoch 1
N <- 500
t <- 200
g[1] <- Ner / N
d[1] <- t / (2 * Ner)

# Epoch 2
N <- 800
t <- 300
g[2] <- Ner / N
d[2] <- t / (2 * Ner)

# Epoch 3
N <- 500
t <- Inf
g[3] <- Ner / N
d[3] <- t / (2 * Ner)

# Obtain the total branch length
longTermBlsL(bls, p1, mu12, mu21, rho, g, d)
```

3.1.2.2 Site frequency spectrum

The definitions of the SFS and the vector $\boldsymbol{g}_h$ are given in Section 3.1.1.2. The following code produces the SFS for the model considered in the previous section.

```
# Create a new object. n1 is the number of alleles associated with A1 in the
    sample.
n <- 10
n1 <- 1
bls <- Sfs2D$new(n, n1)

# Prepare the object
bls$reset()

# Parameters shared across epochs
```

```
p1 <- 0.35
p2 <- 1 - p1
rho <- 2
Ner <- 1000 # Reference population size
M12 <- rho * p2
M21 <- rho * p1

# Epoch 1
N <- 500
t <- 200
g <- c(Ner / (N * p1), Ner / (N * p2))
d <- t / (2 * Ner)
bls$addEpoch(M12, M21, g, d)

# Epoch 2
N <- 800
t <- 300
g <- c(Ner / (N * p1), Ner / (N * p2))
d <- t / (2 * Ner)
bls$addEpoch(M12, M21, g, d)

# Epoch 3
N <- 500
t <- Inf
g <- c(Ner / (N * p1), Ner / (N * p2))
d <- t / (2 * Ner)
bls$addEpoch(M12, M21, g, d)

# Obtain the SFS (i.e., X_i)
bls$getSFS()
```

3.2 A model with recent balancing selection

Consider a diploid panmictic population with constant effective population size N_e. Time is scaled in units of $2N_e$ generations. At an autosomal locus, a mutation from A_1 (the wild type) to A_2 (the mutant) arises. The fitnesses of the genotypes A_1A_1, A_1A_2, and A_2A_2 are $w_{11} = 1 - s_1$, $w_{12} = 1$, and $w_{22} = 1 - s_2$ ($s_1 > 0$ and $s_2 > 0$; i.e., there is heterozygote advantage). We ignore reversible mutation between A_1 and A_2. Define the scaled selection coefficients as $\gamma_1 = 2N_e s_1$ and $\gamma_2 = 2N_e s_2$. Forwards in time, the frequency of A_2 is assumed to increase to $\epsilon = 1/\gamma_1$ instantly after the mutation first arises. This point is taken as the origin of the time axis (i.e., $t = 0$). After that, its frequency increases deterministically, such that the frequency of A_2 at time t is determined by Eq. (23) in Zeng *et al.* (2020). The equilibrium frequencies of A_2 is $\hat{p}_2 = s_1/(s_1 + s_2)$ or equivalently $\gamma_1/(\gamma_1 + \gamma_2)$. The model has an extra parameter δ. Let t_δ be the time when the frequency of A_2 is $\hat{p}_2 - \delta$ (according to the deterministic equation). For $t > t_\delta$, the frequency of A_2 is assumed to be $\hat{p}_2$. Thus, δ controls when the deterministic growth phase ends.

As detailed in Zeng *et al.* (2020), time in the deterministic growth phase is discretised. This is controlled by the parameter H. Specifically, the frequency interval $[\epsilon, \hat{p}_{2,\delta}]$, where $\hat{p}_{2,\delta} = \hat{p}_2 - \delta$, is divided into H equal-sized bins. The harmonic mean of the frequency of

A_2 in each bin is calculated and used to represent the frequency of A_2 in this bin. The results are generally not very sensitive to H, as long as the bins are not very large. In the examples below, we choose H such that each bin represents less than 1% change in allele frequency. This is a very conservative choice and comes at the cost of slower computation. In general, computational complexity increases linearly with H. Further details can be found in the documentation for the R6 classes `RecentBlsModel` and `RecentBlsL`.

3.2.1 Total branch length

Take the model with $\hat{p}_2 = 0.75$ in Figure 6 of Zeng *et al.* (2020) as an example. Here $\gamma_1 = 500$ and $\gamma_2 = \gamma_1/3$.

```
# Set up the selection model
gamma1 <- 500
gamma2 <- gamma1 / 3
delta <- 1e-6
bls <- RecentBlsModel$new(gamma1, gamma2, delta)
# The object bls has a number of functions that may be useful for other
   purposes; see documentation for more details.

# Create a new object for the calculation
n <- 10 # The sample size
blsL <- RecentBlsL$new(n, bls, method = "harmonic", H = 75)
# The selection model can be replaced by calling blsL$setRbls(newBls, method =
    "harmonic", H), where newBls a object created by RecentBlsModel$new

# Obtain the total branch length
rho <- 2 # The scaled recombination rate between the selected and neutral site
t <- 0.04 # Scaled time since the frequency of A2 first reached epsilon
n1 <- 1 # The number of alleles in the sample associated with A1. If it is set
    to -1, then branch lengths for all possible sampling configurations are
   caculated. The sampling configurations are (0, n), (1, n - 1), ..., (n, 0).
blsL$getL(rho, t, n1)

rho <- 10
t <- 0.5
n1 <- -1
blsL$getL(rho, t, n1)
```

3.2.2 Site frequency spectrum

Consider a sample of n alleles at a linked neutral site, with n_1 alleles associated with A_1 and n_2 alleles associated with A_2. Let X_i be the total length of all branches in the genealogy that lead to i alleles in the sample ($0 < i < n$). Evidently $\sum_{i=1}^{n-1} X_i$ is the total branch length. If we define $\theta = 2N_{e,r}u$, where u is the neutral mutation rate per generation. Then under the infinite-sites model, θX_i is the expected number of polymorphic sites in the sample at which the derived allele appears i times. Note that the sample size is limited to $n \leq 12$ (see Section 4 for performance issues). The following example uses the same selection model as the example presented in the previous section.

```
# Set up the selection model
```

```
gamma1 <- 500
gamma2 <- gamma1 / 3
delta <- 1e-6
bls <- RecentBlsModel$new(gamma1, gamma2, delta)

# Create a new object for the calculation
n <- 8 # The sample size
blsSFS <- RecentBlsSFS$new(n, bls, method = "harmonic", H = 75)
# The selection model can be replaced by calling blsSFS$setRbls(newBls, method
     = "harmonic", H), where newBls a object created by RecentBlsModel$new

# Obtain the SFS
rho <- 2 # The scaled recombination rate between the selected and neutral site
t <- 0.04 # Scaled time since the frequency of A2 first reached epsilon
n1 <- 1 # The number of alleles in the sample associated with A1. If it is set
     to -1, then SFSs for all possible sampling configurations are caculated.
    The sampling configurations are (0, n), (1, n - 1), ..., (n, 0).
blsSFS$getSFS(rho, t, n1)

rho <- 10
t <- 0.5
n1 <- -1
sfs <- blsSFS$getSFS(rho, t, n1)
```

3.3 A model with recent positive selection (selective sweep)

Consider a diploid panmictic population with constant effective population size N_e. Time is scaled in units of $2N_e$ generations. At an autosomal locus, a mutation from A_1 (the wild type) to A_2 (the mutant) arises. The fitnesses of the genotypes A_1A_1, A_1A_2, and A_2A_2 are $w_{11} = 1$, $w_{12} = 1 + s$, and $w_{22} = 1 + 2s$ ($s > 0$; i.e., there is semi-dominance). We ignore reversible mutation between A_1 and A_2. Define the scaled selection coefficient as $\gamma = 2N_e s$. Forwards in time, the frequency of A_2 is assumed to increase to $\epsilon = 1/\gamma$ instantly after the mutation first arises. This point is taken as the origin of the time axis (i.e., $t = 0$). After that, its frequency increases deterministically, such that the frequency of A_2 at time t is determined by Eq. (24) in Zeng *et al.* (2020). The deterministic growth phase ends when the frequency of A_2 reaches $1 - \epsilon$. After that, it is regarded as fixed, and evolution is neutral.

As detailed in Zeng *et al.* (2020), time in the deterministic growth phase is discretised. This is controlled by the parameter H. Specifically, the frequency interval $[\epsilon, 1 - \epsilon]$ is divided into H equal-sized bins. The harmonic mean of the frequency of A_2 in each bin is calculated and used to represent the frequency of A_2 in this bin. The results are generally not very sensitive to H, as long as the bins are not very large. In the examples below, we choose H such that each bin represents less than 1% change in allele frequency. This is a very conservative choice and comes at the cost of slower computation. In general, computational complexity increases linearly with H. Further details can be found in the documentation for the R6 classes `RecentSswModel` and `RecentSswL`.

3.3.1 Total branch length

Take the sweep model with $\gamma = 500$ in Figure 6 of Zeng *et al.* (2020) as an example.

```
# Set up the selection model
gamma <- 500
ssw <- RecentSswModel$new(gamma)
# The object ssw has a number of functions that may be useful for other
   purposes; see documentation for more details.
tEnd <- ssw$getTimeUpEpsilon() # When the deterministic growth phase ends.

# Create a new object for the calculation
n <- 10 # The sample size
sswL <- RecentSswL$new(n, ssw, method = "harmonic", H = 100)
# The selection model can be replaced by calling sswL$setRssw(newSsw, method =
    "harmonic", H), where newSsw a object created by RecentSswModel$new

# Obtain the total branch length
rho <- 2 # The scaled recombination rate between the selected and neutral site
t <- tEnd + 0.05 # Sampling after fixation
sswL$getL(rho, t)

rho <- 10
t <- tEnd - 0.01 # Sampling when A2 is still polymorphic
# The number of alleles associated with A1 in the sample. When set to -1, the
   function returns the total branch lengths for all possible sampling
   configurations, ordered as (0, n), (1, n - 1), ..., (n, 0).
n1 <- -1
sswL$getL(rho, t, n1)
```

3.3.2 Site frequency spectrum

Consider a sample of n alleles at a linked neutral site. Let X_i be the total length of all branches in the genealogy that lead to i alleles in the sample ($0 < i < n$). Evidently $\sum_{i=1}^{n-1} X_i$ is the total branch length. If we define $\theta = 2N_e u$, where u is the neutral mutation rate per generation. Then under the infinite-sites model, θX_i is the expected number of polymorphic sites in the sample at which the derived allele appears i times. If the sample is taken before the fixation of A_2, then n_1 alleles in the sample are associated with A_1 and n_2 alleles associated with A_2. Thus, prior to fixation the SFS is dependent on n_1 and n_2. Note that the sample size is limited to $n \leq 12$ (see Section 4 for performance issues). The following example uses the same selection model as the example in the previous section.

```
# Set up the selection model
gamma <- 500
ssw <- RecentSswModel$new(gamma)
tEnd <- ssw$getTimeUpEpsilon() # When the deterministic growth phase ends.

# Create a new object for the calculation
n <- 8 # The sample size
sswSFS <- RecentSswSFS$new(n, ssw, method = "harmonic", H = 100)

# Obtain the SFS (i.e., X_i)
rho <- 2 # The scaled recombination rate between the selected and neutral site
t <- tEnd + 0.05 # Sampling after fixation
sswSFS$getSFS(rho, t)
```

```
rho <- 10
t <- tEnd - 0.01 # Sampling before fixation
n1 <- -1 # The number of alleles in the sample associated with A1. If it is
    set to -1, then branch lengths for all possible sampling configurations are
     caculated. The sampling configurations are (0, n), (1, n - 1), ..., (n, 0)
    .
sswSFS$getSFS(rho, t, n1)
```

3.4 A neutral two-deme model with variable migration rates and/or deme sizes

Consider an island model with two demes. Going backward in time, the history is composed of H discrete epochs. The most recent epoch is epoch 1, and the most ancient epoch has an infinite time span. Let $N_{e,r}$ be a reference diploid population size. In epoch h ($1 \leq h \leq H$), we have the following parameters: (1) $m_{ij,h}$, the (backward) migration rate from deme i to deme j per generation; (2) $N_{i,h}$, the number of breeding individuals in deme i; (3) t_h, the duration of the epoch in generations.

Scale time in units of $2N_{e,r}$ generations. We can define the scaled migration rate as $M_{ij,h} = 2N_{e,r}m_{ij,h}$. We also define $g_h = N_{e,r}/N_{e,h}$, where $N_{e,h} = N_{1,h} + N_{2,h}$. Further, define $p_{1,h} = N_{1,h}/N_{e,h}$ and $d_h = t_h/(2N_{e,r})$. The dynamics in epoch h are determined by $M_{12,h}$, $M_{21,h}$, $p_{1,h}$, g_h, and d_h. Note that either of the two scaled migration rates can be zero, but not both at the same time.

3.4.1 An example equilibrium model

This model has a single epoch of infinite length. Let the reference population size be $N_{e,r} = N_1 + N_2$, where N_i is the number of breeding individuals in deme i.

3.4.1.1 Total branch length

```
# Create a new object. n1 is the number of alleles in deme 1 in the sample.
    See documentation for more details.
n <- 10
n1 <- 1
bls <- BranchLength2D$new(n, n1)

# Prepare the object before adding a new epoch
bls$reset()

p1 <- 0.35 # Relative size of deme 1
M12 <- 0.2
M21 <- 0.3
g <- 1 # Ner / (N1 + N2)
d <- Inf # The equilibrium model is composed of a single epoch of infinite
    length.

# Add the epoch
bls$addEpoch(M12, M21, p1, g, d)
```

```
# Obtain the total branch length (i.e., evaluating aUD in the paper)
bls$getL()
```

It is possible for obtaining the total branch length for all possible sampling configurations simultaneously. This can be done by setting `n1` to -1. Then `bls$getL()` returns a list of values, one for each sampling configuration. See the documentation for `BranchLength2D` for more details.

3.4.1.2 Site frequency spectrum

Consider a sample of n alleles with n_1 alleles in deme 1 and n_2 alleles in deme 2. Let X_i be the total length of all branches in the genealogy that lead to i alleles in the sample ($0 < i < n$). Evidently $\sum_{i=1}^{n-1} X_i$ is the total branch length. If we define $\theta = 2N_{e,r}u$, where u is the neutral mutation rate per generation. Then under the infinite-sites model, θX_i is the expected number of polymorphic sites in the sample at which the derived allele appears i times. Note that the sample size is limited to $n \leq 12$ (see Section 4 for performance issues).

The model is parameterised slightly differently. In epoch h, the definitions of $M_{ij,h}$ and d_h are unchanged. However, instead of having g_h and $p_{1,h}$ as above, we define a vector $\boldsymbol{g}_h = (N_{e,r}/N_{1,h}, N_{e,r}/N_{2,h})$.

The code listed below generates the SFS for the example model considered in the previous section.

```
# Create a new object. n1 is the number of alleles in deme 1 in the sample.
n <- 10
n1 <- 1
bls <- Sfs2D$new(n, n1)

# Prepare the object before adding a new epoch
bls$reset()

M12 <- 0.2
M21 <- 0.3
# p1 = 0.35 and N_ref = N1 + N2 in the previous example
p1 <- 0.35
g <- c(1 / p1, 1 / (1 - p1))
d <- Inf # The equilibrium model is composed of a single epoch of infinite
    length.

# Add the epoch
bls$addEpoch(M12, M21, g, d)

# Obtain the SFS (i.e., X_i)
bls$getSFS()
```

3.4.2 An example model with variable migration rates and deme sizes

The model is displayed in Figure 2. To scale time and the other parameters, we arbitrarily set $N_{e,r} = 1,000$.

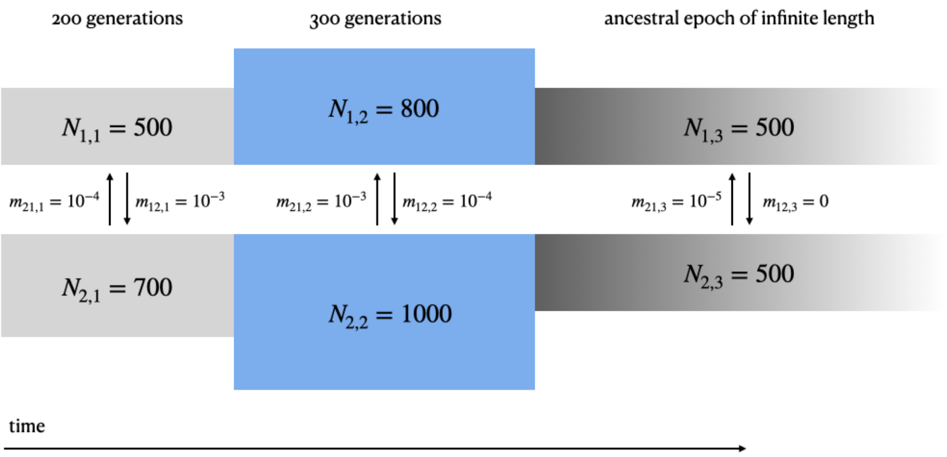

Figure 2: A model with variable migration rates and deme sizes.

3.4.2.1 Total branch length

```
# Create a new object. n1 is the number of alleles in deme 1 in the sample.
   See documentation for more details.
n <- 10
n1 <- 1
bls <- BranchLength2D$new(n, n1)

# Prepare the object before adding a new epoch
bls$reset()

Ner <- 1000 # The reference diploid population size

# Epoch 1
m12 <- 1e-3
m21 <- 1e-4
N1 <- 500
N2 <- 700
t <- 200
M12 <- 2 * Ner * m12
M21 <- 2 * Ner * m21
p1 <- N1 / (N1 + N2)
g <- Ner / (N1 + N2)
d <- t / (2 * Ner)
bls$addEpoch(M12, M21, p1, g, d)

# Epoch 2
m12 <- 1e-4
m21 <- 1e-3
N1 <- 800
N2 <- 1000
t <- 300
M12 <- 2 * Ner * m12
M21 <- 2 * Ner * m21
p1 <- N1 / (N1 + N2)
```

```
g <- Ner / (N1 + N2)
d <- t / (2 * Ner)
bls$addEpoch(M12, M21, p1, g, d)

# Epoch 3
m12 <- 0
m21 <- 1e-5
N1 <- 500
N2 <- 500
t <- Inf
M12 <- 2 * Ner * m12
M21 <- 2 * Ner * m21
p1 <- N1 / (N1 + N2)
g <- Ner / (N1 + N2)
d <- t / (2 * Ner)
bls$addEpoch(M12, M21, p1, g, d)

# Obtain the total branch length (i.e., evaluating aUD in the paper)
bls$getL()
```

3.4.2.2 Site frequency spectrum

Refer to Section 3.4.1.2 regarding the definition of the SFS and the vector $\boldsymbol{g}_h$.

```
# Creates a new object. n1 is the number of alleles in deme 1 in the sample.
n <- 10
n1 <- 1
bls <- Sfs2D$new(n, n1)

# Prepare the object before adding a new epoch
bls$reset()

Ner <- 1000 # The reference diploid population size

# Epoch 1
m12 <- 1e-3
m21 <- 1e-4
N1 <- 500
N2 <- 700
t <- 200
M12 <- 2 * Ner * m12
M21 <- 2 * Ner * m21
g <- c(Ner / N1, Ner / N2)
d <- t / (2 * Ner)
bls$addEpoch(M12, M21, g, d)

# Epoch 2
m12 <- 1e-4
m21 <- 1e-3
N1 <- 800
N2 <- 1000
```

```
t <- 300
M12 <- 2 * Ner * m12
M21 <- 2 * Ner * m21
g <- c(Ner / N1, Ner / N2)
d <- t / (2 * Ner)
bls$addEpoch(M12, M21, g, d)

# Epoch 3
m12 <- 0
m21 <- 1e-5
N1 <- 500
N2 <- 500
t <- Inf
M12 <- 2 * Ner * m12
M21 <- 2 * Ner * m21
g <- c(Ner / N1, Ner / N2)
d <- t / (2 * Ner)
bls$addEpoch(M12, M21, g, d)

# Obtain the SFS (i.e., X_i)
bls$getSFS()
```

3.5 A neutral model with changes in population size

Consider a neutral model with a randomly mating population. Going backward in time, the history is composed of H discrete epochs. The most recent epoch is epoch 1, and the most ancient epoch has an infinite time span. Let $N_{e,r}$ be a reference diploid population size. In epoch h ($1 \leq h \leq H$), we have the following parameters: (1) the population size $N_{e,h}$; (2) t_h, the duration of the epoch in generations. Scale time in units of $2N_{e,r}$ generations. We define $g_h = N_{e,r}/N_{e,h}$ and $d_h = t_h/(2N_{e,r})$. These two parameters determined the dynamics in epoch h. The model has been studied before using different methods (e.g., Marth *et al.*, 2004). The following provides an alternative solution.

3.5.1 An example equilibrium model

This is just the simple neutral model with constant population size. Its properties are well known and are used as a sanity check. Let N_e be the effective population size. Scale time in units of $2N_e$ generations.

3.5.1.1 Total branch length

```
# Create a new object.
n <- 10
bls <- BranchLength1D$new(n)

# Prepare the object before adding a new epoch
bls$reset()

# Add the epoch
g <- 1 # Constant population size
```

```
d <- Inf # The equilibrium model is composed of a single epoch of infinite
    length.
bls$addEpoch(g, d)

# Obtain the total branch length (i.e., evaluating aUD in the paper)
bls$getL()
```

3.5.1.2 Site frequency spectrum

For a sample of n alleles, let X_i be the total length of all branches in the genealogy that lead to i alleles in the sample ($0 < i < n$). Evidently $\sum_{i=1}^{n-1} X_i$ is the total branch length. If we define $\theta = 2N_{e,r}u$, where u is the neutral mutation rate per generation. Then under the infinite-sites model, θX_i is the expected number of polymorphic sites in the sample at which the derived allele appears i times. Note that the sample size is limited to $n \leq 23$ (see Section 4 for performance issues).

```
# Create a new object.
n <- 10
trackAll <- FALSE # Always set to FALSE
bls <-Sfs1D$new(n, trackAll)

# Prepare the object before adding a new epoch
bls$reset()

# Add the epoch
g <- 1 # Constant population size
d <- Inf # The equilibrium model is composed of a single epoch of infinite
    length.
bls$addEpoch(g, d)

# Obtain the SFS (i.e., X_i)
bls$getSFS()
```

3.5.2 An example non-equilibrium model

The model is displayed in Figure 3. To scale time and the other parameters, we arbitrarily set $N_{e,r} = 1,000$.

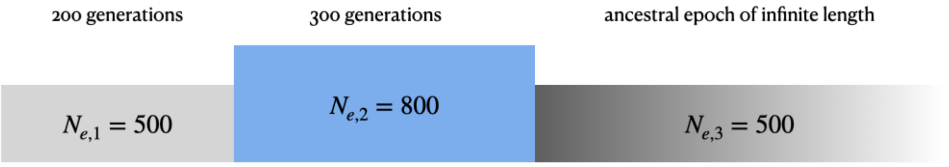

Figure 3: A model with changes in population size.

3.5.2.1 Total branch length

```
# Create a new object.
n <- 10
bls <- BranchLength1D$new(n)

# Prepare the object before adding a new epoch
bls$reset()

Ner <- 1000

# Epoch 1
N <- 500
t <- 200
g <- Ner / N
d <- t / (2 * Ner)
bls$addEpoch(g, d)

# Epoch 2
N <- 800
t <- 300
g <- Ner / N
d <- t / (2 * Ner)
bls$addEpoch(g, d)

# Epoch 3
N <- 500
t <- Inf # Ancestral epoch of infinite length.
g <- Ner / N
d <- t / (2 * Ner)
bls$addEpoch(g, d)

# Obtain the total branch length (i.e., evaluating aUD in the paper)
bls$getL()
```

3.5.2.2 Site frequency spectrum

The SFS is defined in 3.5.1.2.

```
# Create a new object.
n <- 10
trackAll <- FALSE # Always set to FALSE
bls <- Sfs1D$new(n, trackAll)

# Prepare the object before adding a new epoch
bls$reset()

Ner <- 1000

# Epoch 1
N <- 500
t <- 200
g <- Ner / N
```

```
d <- t / (2 * Ner)
bls$addEpoch(g, d)

# Epoch 2
N <- 800
t <- 300
g <- Ner / N
d <- t / (2 * Ner)
bls$addEpoch(g, d)

# Epoch 3
N <- 500
t <- Inf # Ancestral epoch of infinite length.
g <- Ner / N
d <- t / (2 * Ner)
bls$addEpoch(g, d)

# Obtain the SFS (i.e., X_i)
bls$getSFS()
```

3.6 An isolation-with-migration model with variable deme sizes and migration rates

Going backwards, this model has two phase. In phase one, the population is subdivided into two demes connected by gene flow. The dynamics in this phase are similar to those described in Section 3.4, except that there is no ancestral epoch of infinite length. Instead, at certain point the the past, the two demes merged into a randomly mating ancestral population. This is phase two. In this phase, the dynamics are the same as those described in Section 3.5. As in Sections 3.4 and 3.5, time is scaled in units of $2N_{e,r}$, where $N_{e,r}$ is a reference diploid population size.

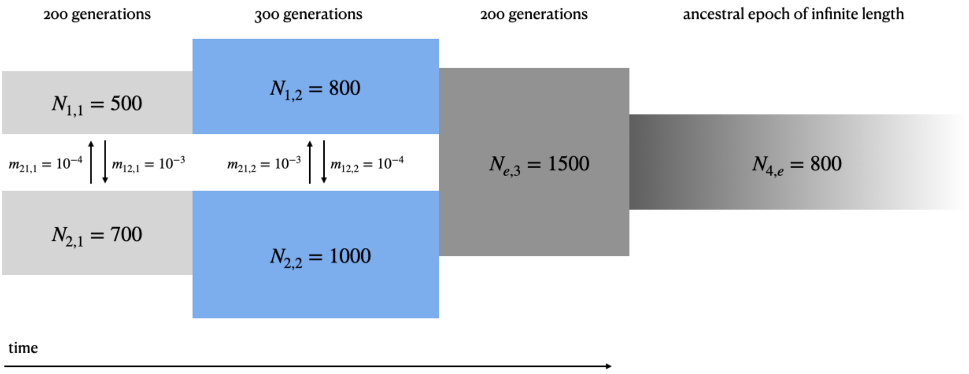

Figure 4: An isolation-with-migration model.

An example is shown in Figure 4. The model has four epochs. The first two contain two demes (phase one). The two demes then merge into an ancestral population (phase two). There are two epochs in phase two. We arbitrarily define $N_{e,r} = 1,000$. We illustrate how to calculate the total branch length and the site frequency spectrum in the follow sections.

3.6.1 Total branch length

```
# Step 1
# Build a BranchLength2D object and add all epochs in phase one, starting from
     the most recent epoch.
n <- 10 # The sample size
n1 <- 1 # The number of alleles in deme 1. If set to -1, the code calculates
    the total branch lengths for all possible sampling configurations, ordered
    as (0, n), (1, n - 1), ..., (n, 0).
bls2d <- BranchLength2D$new(n, n1)

# Prepare the object before adding epochs
bls2d$reset()

Ner <- 1000 # The reference diploid population size

# Epoch 1
m12 <- 1e-3
m21 <- 1e-4
N1 <- 500
N2 <- 700
t <- 200
M12 <- 2 * Ner * m12
M21 <- 2 * Ner * m21
p1 <- N1 / (N1 + N2)
g <- Ner / (N1 + N2)
d <- t / (2 * Ner)
bls2d$addEpoch(M12, M21, p1, g, d)

# Epoch 2
m12 <- 1e-4
m21 <- 1e-3
N1 <- 800
N2 <- 1000
t <- 300
M12 <- 2 * Ner * m12
M21 <- 2 * Ner * m21
p1 <- N1 / (N1 + N2)
g <- Ner / (N1 + N2)
d <- t / (2 * Ner)
bls2d$addEpoch(M12, M21, p1, g, d)

# Step 2
# Build a BranchLength1DFull object and add all epochs in phase one, starting
    from the most recent epoch in this phase.
bls1d <- BranchLength1DFull$new(n)

# Prepare the object before adding epochs
bls1d$reset()

# Epoch 1 in phase two (epoch 3 from the present)
```

```
N <- 1500
t <- 200
g <- Ner / N
d <- t / (2 * Ner)
bls1d$addEpoch(g, d)

# Epoch 2 in phase two (epoch 4 from the present)
N <- 800
t <- Inf
g <- Ner / N
d <- t / (2 * Ner)
bls1d$addEpoch(g, d)

# Step 3
# Pass the two objects to getImL
getImL(bls2d, bls1d)
```

3.6.2 Site frequency spectrum

Consider a sample of n alleles with n_1 alleles in deme 1 and n_2 alleles in deme 2. Let X_i be the total length of all branches in the genealogy that lead to i alleles in the sample ($0 < i < n$). Evidently $\sum_{i=1}^{n-1} X_i$ is the total branch length. If we define $\theta = 2N_{e,r}u$, where u is the neutral mutation rate per generation. Then under the infinite-sites model, θX_i is the expected number of polymorphic sites in the sample at which the derived allele appears i times. Note that the sample size is limited to $n \leq 12$ (see Section 4 for performance issues).

The model is parameterised slightly differently. In epoch h, the definitions of $M_{ij,h}$ and d_h are unchanged. However, instead of having g_h and $p_{1,h}$ as above, we define a vector $\boldsymbol{g}_h = (N_{e,r}/N_{1,h}, N_{e,r}/N_{2,h})$.

```
# Step 1
# Build a Sfs2D object and add all epochs in phase one, starting from the most
    recent epoch.
n <- 10 # The sample size
n1 <- 1 # The number of alleles in deme 1. If set to -1, the SFSs for all
   possible sampling configurations are returned. The sampling configurations
   are (0, n), (1, n - 1), ..., (n, 0).
bls2d <- Sfs2D$new(n, n1)

# Prepare the object before adding epochs
bls2d$reset()

Ner <- 1000 # The reference diploid population size

# Epoch 1
m12 <- 1e-3
m21 <- 1e-4
N1 <- 500
N2 <- 700
t <- 200
```

```
M12 <- 2 * Ner * m12
M21 <- 2 * Ner * m21
g <- c(Ner / N1, Ner / N2)
d <- t / (2 * Ner)
bls2d$addEpoch(M12, M21, g, d)

# Epoch 2
m12 <- 1e-4
m21 <- 1e-3
N1 <- 800
N2 <- 1000
t <- 300
M12 <- 2 * Ner * m12
M21 <- 2 * Ner * m21
g <- c(Ner / N1, Ner / N2)
d <- t / (2 * Ner)
bls2d$addEpoch(M12, M21, g, d)

# Step 2
# Build a Sfs1D object with trackAll set to TRUE and add all epochs in phase
   one, starting from the most recent epoch in this phase.
trackAll <- TRUE
bls1d <- Sfs1D$new(n, trackAll)

# Prepare the object before adding epochs
bls1d$reset()

# Epoch 1 in phase two (epoch 3 from the present)
N <- 1500
t <- 200
g <- Ner / N
d <- t / (2 * Ner)
bls1d$addEpoch(g, d)

# Epoch 2 in phase two (epoch 4 from the present)
N <- 800
t <- Inf
g <- Ner / N
d <- t / (2 * Ner)
bls1d$addEpoch(g, d)

# Step 3
# Pass the two objects to getImSFS
getImSFS(bls2d, bls1d)
```

4 Numerical performance

For numerical efficiency, some of the R code for calculating the SFS was generated using a Java program, which is available upon request. Nonetheless, the sample size the R package can handle is modest. There are several reasons.

1. Numerical linear algebra seems to be rather slow in R. A preliminary comparison to an earlier version of the code developed in `Matlab` has suggested that R may be two orders of magnitude slower! The `Matlab` code is available upon request. It is unfortunately much less user-friendly and does not contain all the functions provided here. We may migrate the R code to Python.

2. The sample size for calculations involving the SFS is particularly restrictive. This is partly because the code has not been optimised by using Eq. (13) in Zeng *et al.* (2020). More importantly, the size of the state space grows very quickly with sample size for these calculations (Andersen *et al.*, 2014), which means that this approach is unlikely to be able to handle sample sizes much larger than the current limit, even after implementing the optimisation.

5 Additional models

Several other models can be implemented relatively easily using the existing framework. These include a two-deme model with population merger and split, an isolation-without-migration model, recent balancing selection and selective sweep models with changes in population size, and a selective sweep model with arbitrary dominance. If any of these are of use, please contact the author.

6 How to cite this package?

Please cite Zeng *et al.* (2020).

7 References

Andersen, L. N., T. Mailund, and A. Hobolth, 2014 Efficient computation in the im model. J Math Biol **68**: 1423–51.

Marth, G. T., E. Czabarka, J. Murvai, and S. T. Sherry, 2004 The allele frequency spectrum in genome-wide human variation data reveals signals of differential demographic history in three large world populations. Genetics **166**: 351–372.

Zeng, K., B. Charlesworth, and A. Hobolth, 2020 Studying models of balancing selection using phase-type theory. bioRxiv **10.1101/2020.07.06.189837**.