The effect on neutral variation of migration and selection at two linked sites

Paths

In[215]:=
```
PlotPath = "/Users/Simon/Documents/LocAdD/results/130419/linkedNeutrSite/figures/"
```

Out[215]=
```
/Users/Simon/Documents/LocAdD/results/130419/linkedNeutrSite/figures/
```

Rules and assumptions

Effective migration rates

For derivations, see *Mathematica* Notebook '2LocContIsland_Determ_effMigRage.nb'.

In[216]:=
```
ruleMeACB := m → m rAC/(a + rAC) rCB/(b + rCB)
ruleMeABC := m → m (b + rAB + rBC)/(a + b + rAB + rBC) rBC/(b + rBC)
ruleMeCAB := m → m rCA/(a + rCA) (a + rCA + rAB)/(a + b + rCA + rAB)
```

Recapitulation of effective migration rates

Formulae for three orderings of loci

The application considered in this *Mathematica* Notebook assumes the fate of a neutral locus linked to a (set of) selected one(s) is largely determined by the effective migration rate experienced by the neutral locus. This effective migration rate depends on the ordering of the loci. Assuming two diallelic selected loci, $\mathcal{A}$ and $\mathcal{B}$, and one neutral locus, C, there are three orderings of interest: $\mathcal{A}$-C-$\mathcal{B}$, $\mathcal{A}$-$\mathcal{B}$-C and C-$\mathcal{A}$-$\mathcal{B}$.

In *Mathematica* Notebook '2LocContIsland_Determ_effMigRate.nb', we derived the effective migration rate for these three constellations (the one for $\mathcal{A}$-C-$\mathcal{B}$ had been previously derived by Bürger and Akerman, 2011). In doing so, we assumed continuous-time dynamics, and in particular that no more than one recombination event happened per time step. Without giving details, we just state the solutions here:

Constellation $\mathcal{A}$-C-$\mathcal{B}$:

$$m_e^{(\mathrm{ACB})} = m \frac{r_{\mathrm{AC}}}{a + r_{\mathrm{AC}}} \frac{r_{\mathrm{CB}}}{b + r_{\mathrm{CB}}}, \tag{1}$$

where r_{AC} and r_{CB} are the recombination rates between loci $\mathcal{A}$ and C, and C and $\mathcal{B}$, respectively, and a and b are the selection coefficients in favour of alleles A_1 and A_2, respectively.

Constellation $\mathcal{A}$-$\mathcal{B}$-C:

$$m_e^{(\mathrm{ABC})} = m \frac{b + r_{\mathrm{AB}} + r_{\mathrm{BC}}}{a + b + r_{\mathrm{AB}} + r_{\mathrm{BC}}} \frac{r_{\mathrm{BC}}}{b + r_{\mathrm{BC}}}, \tag{2}$$

where r_{AB} and r_{BC} are the recombination rates between loci $\mathcal{A}$ and $\mathcal{B}$, and $\mathcal{B}$ and C, respectively.

Constellation C-$\mathcal{A}$-$\mathcal{B}$:

$$m_e^{(\mathrm{CAB})} = m\,\frac{r_{\mathrm{CA}}}{a + r_{\mathrm{CA}}}\,\frac{a + r_{\mathrm{CA}} + r_{\mathrm{AB}}}{a + b + r_{\mathrm{CA}} + r_{\mathrm{AB}}}, \tag{3}$$

with obvious definitions of r_{CA} and r_{AB}.

A comment on the validity of the effective migration rates

Importantly, these effective migration rates were derived under the assumption that either migration is much weaker compared to selection and recombination, *i.e.* $m \ll \min(a, r_{\mathrm{AB}})$, or that recombination is much weaker compared to migration and selection, *i.e.* $r_{\mathrm{AB}} \ll \min(m, a)$.

Under the assumption of weak migration, the coordinates of the fully-polymorphic asymptotically stable two-locus migration-selection equilibrium $E_+$are, up to first order of m, given by

$$\hat{p}_+ = 1 - \frac{m}{a}\left(1 - \frac{b}{a + b + r_{\mathrm{AB}}}\right) + O\left(m^2\right) \tag{4}$$

$$\hat{q}_+ = 1 - \frac{m}{b}\left(1 - \frac{a}{a + b + r_{\mathrm{AB}}}\right) + O\left(m^2\right) \tag{5}$$

$$\hat{D}_{\mathrm{AB}+} = \frac{m}{a + b + r_{\mathrm{AB}}} + O\left(m^2\right) \tag{6}$$

These are immediately obtained from formulae given by Bürger and Akerman (2011, Eq. 4.1). Moreover, at this equilibrium, the additional coordinates for a model with a third neutral locus are given by $\hat{n} = n_c$ and $\hat{D}_{\mathrm{AC}} = \hat{D}_{\mathrm{CB}} = \hat{D}_{\mathrm{ACB}} = 0$.

Under the assumption of tight linkage (weak recombination), the equilibrium coordinates for p, q and D_{AB} are

$$\hat{p}_+ = 1 - \frac{m}{a + b} - \frac{r_{\mathrm{AB}}\,m}{(a + b)^2}\left[\frac{b}{a} - \frac{m}{a + b}\left(\frac{b}{a} - 1\right)\right] + O\left(r^2\right) \tag{7}$$

$$\hat{q}_+ = 1 - \frac{m}{a + b} - \frac{r_{\mathrm{AB}}\,m}{(a + b)^2}\left[\frac{a}{b} + \frac{m}{a + b}\left(1 - \frac{a}{b}\right)\right] + O\left(r^2\right) \tag{8}$$

$$\hat{D}_{\mathrm{AB}+} = \frac{m}{a + b}\left(1 - \frac{m}{a + b}\right) - \frac{r_{\mathrm{AB}}\,m}{(a + b)^2}\left[1 - \frac{m}{a + b}\left(1 - \frac{m}{a + b}\right)\left(2 - \frac{b}{a} - \frac{a}{b}\right)\right] + O\left(r_{\mathrm{AB}}{}^2\right) \tag{9}$$

and those for n, D_{AC}, D_{CB} and D_{ACB} are as above.

Absorption (with $n_c = 0$) and stationary distribution (with $n_c > 0$)

Diffusion approximation for the neutral one-locus migration model

We first assume a neutral one-locus model with continent-island migration and recapitulate basic diffusion theory.

Call the two alleles C_1 and $C_2$and let n and n_c be the frequency of C_1 on the island and continent, respectively. We assume that n_c is constant. This would, for instance, correspond to a case where C_1 is segregating at mutation-drift balance on the continent. In principle, mutation could also happen on the island, but if the island population is of moderate size, then the mutational input is limited and – in the absence of migration – the variation at any site determined mainly by genetic drift. Further, assume that a proportion m of the island population is replaced by migrants from the continent each generation. Note that immigration is the force that keeps the neutral site polymorphic.

Recursion equation in discrete time

Given the assumptions stated above, we obtain the recursion equation for n as

$$n' = (1 - m)\,n \;+\; m\,n_c. \tag{10}$$

Differential equations in continuous time

We derive the continuous-time version from the difference equation

$$\Delta \mathrm{n} \;=\; (1 - m)\,n \;+\; m\,n_c - n = m(n_c - n), \tag{11}$$

the right-hand side of which in this case directly corresponds to the differential, *i.e.*

$$\dot{n} := \frac{d\,n}{d\,t} = m\,(n_c - n). \tag{12}$$

We note that allele C_1 is doomed to extinction whenever $n_c = 0$.

■ Equations for the mean and variance of the change in δt

On the diffusion scale ($\mu = 2\, N_e\, m$), the mean and variance of the change in n over a small amount of time δt are given by

$$M(n) = \mu(\, n_c - n) \tag{13}$$

and

$$V(n) \;= n(1 - n), \tag{14}$$

respectively, where time is measured in units of $2\, N_e$.

In[219]:=
```
MOLMN[n_] := μ (nC - n)
VOLMN[n_] := n (1 - n)
```

■ A comment to the distinction between $n_c = 0$ vs. $n_c > 0$

There is an important difference between the cases of $n_c = 0$ and $n_c > 0$. In the former, $n = 0$ is an absorbing state under the one-way migration scheme of the continent-island model, and therefore C_1will go extinct with probability 1. It makes sense to compute the sojourn-time density and the mean absorption time, but invasion is not possible in the long term. In the case of $n_c > 0$, however, allele C_1 is introduced at a constant rate to the island via gene flow, such that C_1 cannot go extinct. Excluding the non-generic case of $n_c = 1$, there is no absorbing state, but there instead exist a stationary distribution of allele frequencies. In the following, we treat these two cases separately.

■ Sojourn-time density and mean absorption time for $n_c = 0$

■ Preliminaries

We consider the where C_1 is absent from the continent, but present on the island. A biological scenario of interest would be the ocurrence of C_1 as a new mutation on the island, in which case 'allele' C_2would subsume all the other alleles segregating at the neutral locus of interest. Although any such novel mutation is doomed to extinction, if there are many of them, over long periods of time, the process may lead to a certain level of neutral differentiation between the continent and the island. This level is expected to be higher compared to the case without linkage to selected loci.

We follow standard diffusion theory based on the backward Kolmogorov equation, as outlined in detail by Ewens (1979).

```
MOLMN[n]
VOLMN[n]
```

$$\frac{(-n + nC)\,\mu}{(1 - n)\,n}$$

```
MOLMN[n]
VOLMN[n] /. {nC → 0}
```

$$-\frac{\mu}{1 - n}$$

```
-2 Integrate[MOLMN[z]
VOLMN[z] /. {nC → 0}, {z, 0, y}, Assumptions → {0 < μ, 0 ≤ nC ≤ 1, 0 < y ≤ 1}]
```

$$-2\,\mu\,\text{Log}[1 - y]$$

```
e^%
// Simplify
```

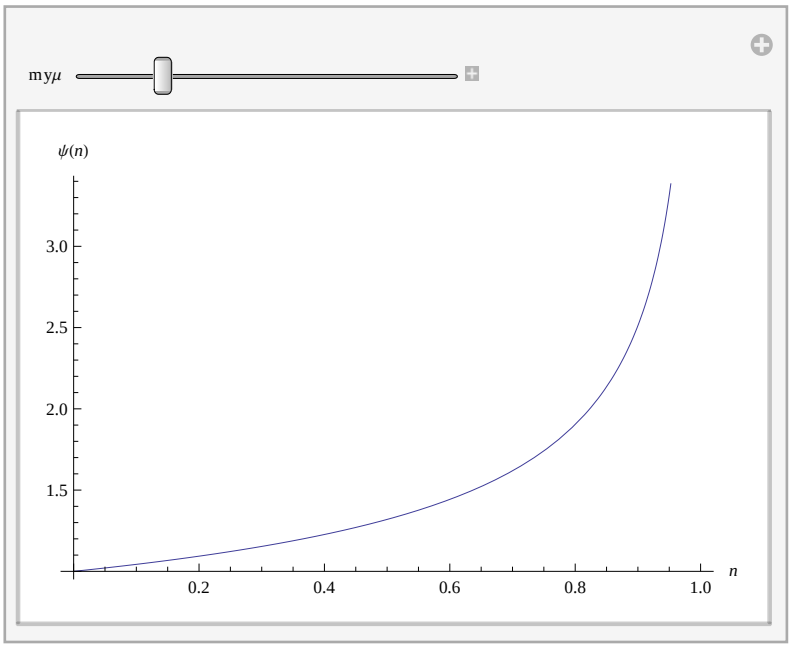
$$(1 - y)^{-2\,\mu}$$

We define the function ψ(y) according to equation (4.16) in Ewens (1979):

$$\psi\,(y) = e^{-2\int_0^y \frac{M(z)}{V(z)}\,dz} = (1 - y)^{-2\,\mu} \tag{15}$$

In[221]:=
```
ψOLMNMonom[y_] := (1 - y)^-2 μ
```

```
Manipulate[Plot[{ψOLMNMonom[y] /. {μ → myμ}},
  {y, 0, 1}, AxesLabel → {n, "ψ(n)"}], {{myμ, 0.2}, 0, 1}]
```



To compute the functions $t(p; p_0)$ given in equations (4.38) and (4.39) in Ewens (1979) [where x and p are used instead of n and n_0], we need the definite integral of $\psi(y)$ defined above, *i.e.* $\int_0^n \psi(y)\, dy$. This is equal to $\frac{1-(1-n)^{1-2\mu}}{1-2\mu}$.

```
FullSimplify[Integrate[ψOLMNMonom[y], {y, 0, n}], Assumptions → {0 ≤ n ≤ 1}]
```

$$\frac{-1+(1-n)^{1-2\mu}}{-1+2\mu}$$

In[222]:=
```
ψOLMNMonomInt[μ_, n_] := (1 - (1 - n)^(1 - 2 μ))/(1 - 2 μ)
```

We consider the ratio of $\frac{1}{n}\int_0^n \psi(y)\, dy$ and investigate if it converges to 1 as $n \to 0$.

```
Limit[ψOLMNMonomInt[μ, n] / n, n → 0]
```

```
1
```

This observation will allow us to simplify expressions further below.

- Diffusion approximation to the sojourn-time density

We turn to the sojourn-times according to Ewens (1979; pp. 141–144). We denote the initial frequency on the island of by n_0. Below, $t_{1,\text{neut}}(n; n_0)$ applies if $0 \le n < n_0$ and $t_{2,\text{neut}}(n; n_0)$ applies if $n_0 \le n \le 1$.

In[223]:=
```
t1OLMN[μ_, n_] := 2/(VOLMN[n] ψOLMNMonomFunc[μ, n]) ψOLMNMonomInt[μ, n] // Simplify
t2OLMN[μ_, n_, n0_] :=
  2/(VOLMN[n] ψOLMNMonomFunc[μ, n]) ψOLMNMonomInt[μ, n0] // FullSimplify
```

```
{t1OLMN[μ, n], t2OLMN[μ, n]} // TableForm
```

$$\frac{2\left(-1+(1-n)^{2\mu}+n\right)}{(-1+n)\, n\, (-1+2\mu)}$$

```
t2OLMN[μ, n]
```

We have

$$t_{1,\text{neut}}(n) = -2\,\frac{1-n-(1-n)^{2\mu}}{n(1-n)(1-2\mu)} = 2\,\frac{(1-n)^{2\mu-1}-1}{n(1-2\mu)} \tag{16}$$

and

$$t_{2,\text{neut}}(n;\,n_0) = 2\,\frac{(1-n)^{2\mu-1}\,(1-(1-n_0)^{1-2\mu})}{n(1-2\mu)}. \tag{17}$$

```
-2 (1 - n - (1 - n)^(2 μ))/(n (1 - n) (1 - 2 μ)) - (2 ((1 - n)^(2 μ - 1) - 1)/(n (1 - 2 μ))) // Simplify
```

```
0
```

```
2/(VOLMN[n] ψOLMNMonomFunc[μ, n]) ψOLMNMonomInt[μ, n] // Simplify
```

$$\frac{2\left(-1+(1-n)^{2\mu}+n\right)}{(-1+n)\,n\,(-1+2\mu)}$$

```
2/(VOLMN[n] ψOLMNMonomFunc[μ, n])
```

$$\frac{2\,(1-n)^{-1+2\mu}}{n}$$

We note that $t_{1,,\text{neut}} = \frac{2\,(1-n)^{2\mu-1}}{n}\int_0^n \psi(y)\,dy = 2\,(1-n)^{2\mu-1}\,\frac{1}{n}\int_0^n \psi(y)\,dy$. We have already seen that $\frac{1}{n}\int_0^n \psi(y)\,dy \to 1$ as $n \to 0$, and we can therefore approximate $t_{1,\text{neut}}(n)$ by $\tilde{t}_{1,\text{neut}}(n) = 2\,(1-n)^{2\mu-1}$ whenever n is small. Recall that $t_{1,\text{neut}}(n)$ is used only if $0 \le n \le n_0$ holds. When considering a single de-novo mutation, we are interested in the case where $n_0 = \frac{1}{2N}$, with N usually of order 100 or larger, ans so $n \le n_0$ automatically implies that n is small whenever $t_{1,\text{neut}}(n)$ is employed. Therefore, we expect the approximation to be valid for a single initial copy of C_1.

In[225]:=
```
t1OLMNApprox[μ_, n_] := 2 (1 - n)^(2 μ - 1)
```

Similarly, we multiply $t_{2,\text{neut}}(n;\,n_0)$ by n_0 and $\frac{1}{n_0}$, which results in no net change, but allows us to write $t_{2,\text{neut}} = \frac{2\,n_0\,(1-n)^{2\mu-1}}{n} \times \frac{1}{n_0}\int_0^{n_0} \psi(y)\,dy$. We have already seen that the second part converges to 1 as $n_0 \to 0$, and we can therefore approximate $t_{2,\text{neut}}(n;\,n_0)$ by $\tilde{t}_{2,\text{neut}}(n;\,n_0) = 2\,\frac{n_0(1-n)^{2\mu-1}}{n}$ whenever n_0 is small.

In[226]:=
```
t2OLMNApprox[μ_, n_, n0_] := 2 n0 (1 - n)^(2 μ - 1)/n
```

In summary, for n_0 small, $t_{1,\text{neut}}(n)$ and $t_{2,\text{neut}}(n;\,n_0)$ are approximated by

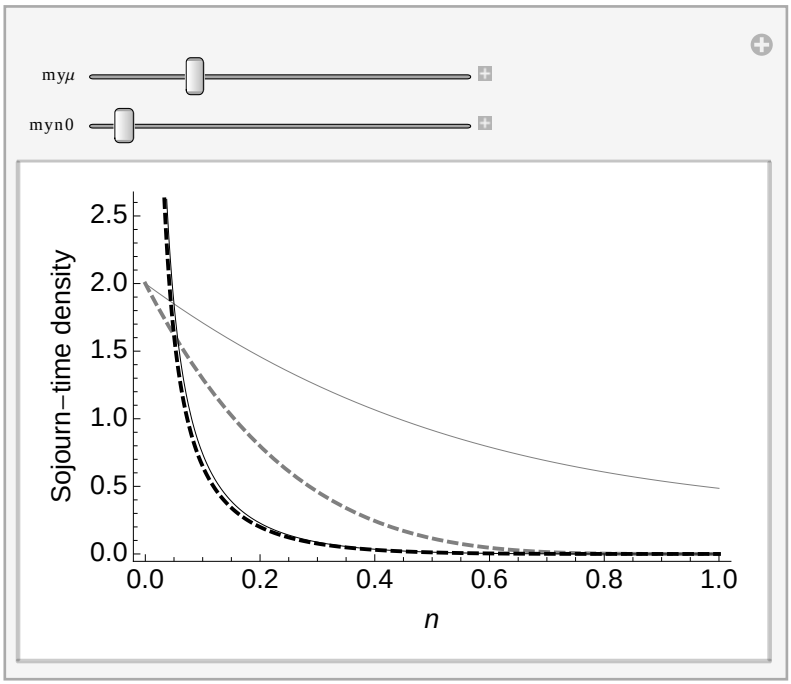
$$\tilde{t}_{1,\text{neut}}(n) = 2\,(1-n)^{2\mu-1} \tag{18}$$

and

$$\tilde{t}_{2,\text{neut}}(n,\,n_0) = 2\,(1-n)^{2\mu-1}\,\frac{n_0}{n}, \tag{19}$$

respectively.

```
Manipulate[Plot[{t1OLMN[μ, n] /. {μ → myμ},
    t1OLMNApprox[μ, n] /. {μ → myμ}, t2OLMN[μ, n, n0] /. {μ → myμ, n0 → myn0},
    t2OLMNApprox[μ, n, n0] /. {μ → myμ, n0 → myn0}}, {n, 0, 1},
  Frame → True, FrameStyle → {{Black, Opacity[0]}, {Black, Opacity[0]}},
  FrameLabel → {n, "Sojourn-time density"},
  LabelStyle → {Directive[FontSize → 14], FontFamily → "Helvetica"},
  PlotStyle → {Gray, {Gray, Dashed, Thick}, Black, {Black, Thick, Dashed}}],
 {{myμ, 0.8}, 0, 10}, {{myn0, 0.05}, 0, 1}]
```



- Diffusion approximation to the mean absorption time

We obtain the mean absorption time following Ewens (1979, pp. 140–145) as

$$\bar{t}_{\text{neut}}(n_0) = \int_0^{n_0} t_{1,\text{neut}}(n)\, dn + \int_{n_0}^{1} t_{2,\text{neut}}(n; n_0)\, dn. \tag{20}$$

```
t1OLMN[μ, n]
```

$$\frac{2\left(-1+(1-n)^{2\mu}+n\right)}{(-1+n)\,n\,(-1+2\mu)}$$

```
t2OLMN[μ, n]
```

```
t2OLMN[μ, n]
```

```
FullSimplify[Integrate[t1OLMN[2 Ne m, n], {n, 0, n0}] + Integrate[t2OLMN[2 Ne m, n, n0],
   {n, n0, 1}, Assumptions → {0 < Ne, 0 < m, 0 < n0 < 1, n0 ∈ Reals}],
 Assumptions → {n0 ∈ Reals, 0 < n0 < 1}]
```

$$\frac{1}{-1+4\,m\,Ne}\,2\,(1-n0)^{-4\,m\,Ne}\left(-(-1+n0)\,n0\,(-1+4\,m\,Ne)\,\text{HypergeometricPFQ}[\{1,1,2-4\,m\,Ne\},\{2,2\},n0]+\left(-1+(1-n0)^{4\,m\,Ne}+n0\right)(\text{EulerGamma}+\text{Log}[n0]+\text{PolyGamma}[0,4\,m\,Ne])\right)$$

```
PolyGamma[0, a] - PolyGamma[a]
```

```
0
```

The mean absorption time can be expressed analytically and is given by

$$\tilde{t}_{\text{neut}}(n_0) = 2\,\frac{1}{1-4\,N_e\,m}\,(1-n_0)^{-4\,m\,N_e}$$
$$\left\{(1-n_0)\,(1-4\,N_e\,m)\,n_0 \times {}_3F_2[(1,\,1,\,2-4\,N_e\,m);\,(2,\,2);\,n_0] + \left[1-(1-n_0)^{4\,N_e\,m}-n_0\right]\left(\gamma + \log(n_0) + \frac{\Gamma'(2\,N_e\,m)}{\Gamma(4\,N_e\,m)}\right)\right\}, \tag{21}$$

where ${}_pF_q$ (a; b; z) is the generalised hypergeometric function (with a = $\{a_1 \ldots a_p\}$ and b = $\{b_1 \ldots b_q\}$), γ is Euler's gamma constant (≈ 0.5772), and $\Gamma(x)$ is the gamma function. The prime denotes the first derivative.

In[227]:=
```
tAbsorbOLMN[m_, Ne_, n0_] := 2 n0 HypergeometricPFQ[{1, 1, 2 - 4 m Ne}, {2, 2}, n0] +
   1/(-1 + 4 m Ne) 2 (1 - n0)^(-4 m Ne) (-1 + (1 - n0)^(4 m Ne) + n0) (EulerGamma + (n0 - 4 m n0 Ne)
     HypergeometricPFQ[{1, 1, 2 - 4 m Ne}, {2, 2}, n0] + Log[n0] + PolyGamma[0, 4 m Ne])
```

When assuming n_0 small, however, this terms can be simplified considerably. Specifically, we recall that $t_{1,\text{neut}}(n)$ and $t_{2,\text{neut}}(n;\,n_0)$ are approximately $\tilde{t}_{1,\text{neut}}(n) = 2\,(1-n)^{2\,\mu-1}$ and $\tilde{t}_{2,\text{neut}}(n;\,n_0) = 2\,\frac{n_0}{n}\,(1-n)^{2\,\mu-1}$, respectively.

This yields, as an approximation to the mean absorption time,

$$\tilde{\tilde{t}}_{\text{neut}}(n_0) = \int_0^{n_0} \tilde{t}_{1,\text{neut}}(n)\,dn + \int_{n_0}^{1} \tilde{t}_{2,\text{neut}}(n;\,n_0)\,dn = 2\int_0^{n_0} (1-n)^{2\,\mu-1}\,dn + 2\,n_0\int_{n_0}^{1} \frac{1}{n}\,(1-n)^{2\,\mu-1}\,dn. \tag{22}$$

```
Integrate[t1OLMNApprox[μ, n], {n, 0, n0}, Assumptions → {0 < n0 < 1, n0 ∈ Reals}]
```

$$\frac{1-(1-n0)^{2\,\mu}}{\mu}$$

```
Integrate[t2OLMNApprox[μ, n, n0], {n, n0, 1}, Assumptions → {0 < n0 < 1, n0 ∈ Reals, 0 < μ}]
```

```
2 n0 (-HarmonicNumber[-1 + 2 μ] +
   n0 (-1 + 2 μ) HypergeometricPFQ[{1, 1, 2 - 2 μ}, {2, 2}, n0] - Log[n0])
```

```
FullSimplify[
 Integrate[t1OLMNApprox[2 Ne m, n], {n, 0, n0}] + Integrate[t2OLMNApprox[2 Ne m, n, n0],
   {n, n0, 1}, Assumptions → {0 < Ne, 0 < m, 0 < n0 < 1, n0 ∈ Reals}],
 Assumptions → {n0 ∈ Reals, 0 < n0 < 1}]
```

$$-\frac{-1+(1-n0)^{4\,m\,Ne}}{2\,m\,Ne} + 2\,n0\,(-\text{HarmonicNumber}[-1+4\,m\,Ne] + n0\,(-1+4\,m\,Ne)\,\text{HypergeometricPFQ}[\{1,1,2-4\,m\,Ne\},\{2,2\},n0] - \text{Log}[n0])$$

Thus, plugging (18) and (19) into Eq. (22) yields the approximate mean absorption time as

$$\tilde{\tilde{t}}_{\text{neut}}(n_0) = -\frac{(1-n_0)^{4\,N_e\,m}-1}{2\,N_e\,m} - 2\,n_0,\,\left[H_n + n_0(1-4\,N_e\,m)\,{}_pF_q((1,\,1,\,2-4\,N_e\,m);\,(2,\,2);\,n_0) + \log(n_0)\right], \tag{23}$$

where $H_n = \sum_{k=1}^{n} \frac{1}{k}$ is the n^{th} harmonic number and ${}_pF_q$ (a; b; z) is the generalised hypergeometric function.

Moreover, assuming $n_0 = \frac{1}{2\,N} = \frac{1}{2\,N_e}$, and noting that then the contribution of t_1 can be ignored, Eq. (22) simplifies to

$$\tilde{\tilde{t}}_{\text{neut}}(n_0) \approx \int_{\frac{1}{2N}}^{1} \tilde{t}_{2,\text{neut}}\left(n;\,\frac{1}{2\,N}\right)dn \overset{N_e=N}{=} \frac{1}{N_e}\int_{\frac{1}{2N}}^{1} n^{-1}(1-n)^{2\,\mu-1}\,dn \overset{\mu=2\,N_e\,m}{=} \frac{1}{N_e}\int_{\frac{1}{2N}}^{1} n^{-1}(1-n)^{4\,N_e\,m-1}\,dn. \tag{24}$$

Note that all times are given on the diffusion scale; to obtain times on the natural scale, terms given here must be multiplied by $2\,N_e$.

See also Ewens (1979, pp. 171–175), where the case of one-way mutation is treated, which is equivalent to one-way migration. Specifically, $2\,\mu = 4\,N_e\,m$ here corresponds to $\theta = 2\,N_e\,u$ in Ewens (1979), where u is the mutation rate from focal allele to the alternative allele(s).

In[228]:=
```
tAbsOLMNApprox[m_, Ne_, n_] := 1/Ne Integrate[n^-1 (1 - n)^(4 Ne m - 1), {n, 1/(2 Ne), 1}]
```

```
Series[n^-1 (1 - n)^(2 μ-1), {μ, 0, 1}] // Normal // FullSimplify
```

$$\frac{1 + 2\,\mu\,\text{Log}[1 - n]}{n - n^2}$$

```
Integrate[1/n, {n, 1/(2 Ne), 1}, Assumptions → {1 < Ne, Ne ∈ Reals}]
```

$$\text{Log}[2\,Ne]$$

```
Integrate[1/(1 - n), {n, 1/(2 Ne), 1}, Assumptions → {1 < Ne, Ne ∈ Reals}]
```

Integrate::idiv : Integral of $\frac{1}{1-n}$ does not converge on $\left\{\frac{1}{2\,Ne}, 1\right\}$. ≫

```
Integrate[1/(1 - n), {n, 1/(2 Ne), 1}, Assumptions → {1 < Ne, Ne ∈ Reals}]
```

```
Integrate[(1 + 2 μ Log[1 - n])/(n - n^2), {n, 1/(2 Ne), 1}, Assumptions → {1 < Ne, Ne ∈ Reals}]
```

Integrate::idiv : Integral of $\frac{1}{n-n^2} + \frac{2\,\mu\,\text{Log}[1-n]}{n-n^2}$ does not converge on $\left\{\frac{1}{2\,Ne}, 1\right\}$. ≫

```
Integrate[(1 + 2 μ Log[1 - n])/(n - n^2), {n, 1/(2 Ne), 1}, Assumptions → {1 < Ne, Ne ∈ Reals}]
```

```
Integrate[(1 + 2 μ Log[1 - n])/n, {n, 1/(2 Ne), 1}, Assumptions → {1 < Ne, Ne ∈ Reals}]
```

$$-\frac{\pi^2\,\mu}{3} + \text{Log}[2\,Ne] + 2\,\mu\,\text{PolyLog}\left[2, \frac{1}{2\,Ne}\right]$$

```
Series[1/Ne Integrate[n^-1 (1 - n)^(2 μ-1), {n, 1/(2 Ne), 1}], {μ, 0, 1}] // Normal // FullSimplify
```

$$\frac{1}{2\,Ne^2}\left((-1 + 2\,\mu)\,\text{HypergeometricPFQ}\left[\{1, 1, 2 - 2\,\mu\}, \{2, 2\}, \frac{1}{2\,Ne}\right] - 2\,Ne\,(\text{EulerGamma} - \text{Log}[2\,Ne] + \text{PolyGamma}[0, 2\,\mu])\right)$$

For moderate $2\,\mu = 4\,N_e\,m$, i.e. $2\,\mu \approx 1$, Eq. (24) is of order $\log(2\,N_e)$, meaning that a new mutation C_1 will, on average, not remain in the island population for a long time.

In[229]:=
```
tAbsOLMNIntegrand[m_, Ne_, n_] := 1/Ne n^-1 (1 - n)^(4 Ne m-1)
```

- Application: insertion of effective migration rates

Insertion of equations (1) to (3) into Eq. (24) yields the approximate mean absorption times for constellations $\mathcal{A}$-$\mathcal{C}$-$\mathcal{B}$, $\mathcal{A}$-$\mathcal{B}$-$\mathcal{C}$ and $\mathcal{C}$-$\mathcal{A}$-$\mathcal{B}$, respectively. The resulting terms cannot be algebraically simplified any further, and so we focus on a graphical exploration.

First, we define a function that returns the mean absorption time as a function of locus $\mathcal{C}$, for fixed positions of loci $\mathcal{A}$ and $\mathcal{B}$.

```
FullSimplify[tAbsOLMNApprox[m, Ne, n], Assumptions → {0 < m, 1 ≤ Ne}]
```

$$\frac{1}{2\,Ne^2}\left((-1 + 4\,m\,Ne)\,\text{HypergeometricPFQ}\left[\{1, 1, 2 - 4\,m\,Ne\}, \{2, 2\}, \frac{1}{2\,Ne}\right] + Ne\left(\text{Log}\left[4\,Ne^2\right] - 2\,(\text{EulerGamma} + \text{PolyGamma}[0, 4\,m\,Ne])\right)\right)$$

```
tAbsOLMNIntegrand[m, Ne, n] /. ruleMeACB
```

$$\frac{(1 - n)^{-1 + \frac{4\,m\,Ne\,rAC\,rCB}{(a + rAC)\,(b + rCB)}}}{n\,Ne}$$

```
tAbsOLMNIntegrand[m, Ne, n] /. ruleMeABC
```

$$\frac{(1-n)^{-1+\frac{4\,m\,Ne\,rBC\,(b+rAB+rBC)}{(b+rBC)\,(a+b+rAB+rBC)}}}{n\,Ne}$$

```
tAbsOLMNIntegrand[m, Ne, n] /. ruleMeCAB
```

$$\frac{(1-n)^{-1+\frac{4\,Ne\,rCA\,(a+rAB+rCA)}{(a+rCA)\,(a+b+rAB+rCA)}}}{n\,Ne}$$

In[230]:=

```
meanAbsTimeFunc::usage =
  "meanAbsTimeFunc[a, b, m, Ne, xA, xB, xC, scaleFac, ratio] returns the mean
    absorption time of a neutral de-novo mutation that is linked to two
    sites under selection, valid for weak migration compared to selection.
    The selection coefficients at the two sites are a and b, the actual
    migration rate is m and the positions of the three sites in map units
    are xA, xB and xC, where the latter belongs to the neutral locus and
    xA < xB is assumed without loss of generality. The scaleFac defines
    what recombination rate should correspond to one map unit, and ratio
    denotes by how much migration must be weaker compared to recombination.";
meanAbsTimeFunc[a_, b_, m_, Ne_, xA_, xB_, xC_, scaleFac_, ratio_] :=
 Module[{me, tAbsorb, rAC, rBC, x},
  rAC = Abs[xA - xC] scaleFac;
  rBC = Abs[xB - xC] scaleFac;
  x = Min[rAC, rBC];
  me = If[xC < xA < xB, m rAC (a + rBC)/((a + rAC) (a + b + rBC)), If[xA ≤ xC ≤ xB,
      m rAC rBC/((a + rAC) (b + rBC)), If[xA < xB < xC, m rBC (b + rAC)/((b + rBC) (a + b + rAC)), Null]]];
  tAbsorb = 1/Ne NIntegrate[n^-1 (1 - n)^(4 Ne me - 1), {n, 1/(2 Ne), 1},
     MinRecursion → 2, MaxRecursion → 15];
  Return[If[xA == xC || xB == xC, Null, If[(m ratio ≤ x), Chop[tAbsorb], Null]]]
  (*Return[If[xA==xC||xB==xC,Null,
    If[(m ratio<a&&m ratio<x )||(x ratio<a&&x ratio<m),tAbs,Null]]]*)
  (*Return[If[a/m<ratio||a/x<ratio||xA==xC||xB==xC,Null,tAbs]]*)
  (* We return 'Null' if xC is so close to xA or xB that we
   cannot justify migration being weak compard to recombination. *)
 ]
```

In[232]:=

```
meanAbsTimeExactFunc::usage="meanAbsTimeExactFunc[a, b, m, Ne, xA, xB, xC, scaleFac, r
meanAbsTimeExactFunc[a_,b_,m_,Ne_,xA_,xB_,xC_,scaleFac_]:=Module[{me,tAbsorb,rAC,rBC,x
rAC=Abs[xA-xC]scaleFac;
rBC=Abs[xB-xC]scaleFac;
JNACB={{-m,a,b,0},{1/(8 a (r1+r2)) m (-a^2+b^2+6 a (r1+r2)-4 m (r1+r2)-(r1+r2)^2+(a-b+r1+r2) √
JNABC={{-m,a,b,0},{m (-a^2+b^2+6 a r1-4 m r1-r1^2+(a-b+r1) √(-8 m r1+(a+b+r1)^2))/(8 a r1),a-m-r1-
JNCAB={{-m,b,a,0},{m (a^2-b^2+6 b r2-4 m r2-r2^2+(-a+b+r2) √(-8 m r2+(a+b+r2)^2))/(8 b r2),b-m-r1-r2
me=If[xC<xA<xB,-Max[Re[Eigenvalues[Chop[JNCAB]]]],If[xA≤xC≤xB,-Max[Re[Eigenvalues[Chop
tAbsorb=1/Ne NIntegrate[n^-1 (1-n)^(4Ne me-1),{n,1/(2Ne),1},MinRecursion→2,MaxRecursion→15];
Return[If[xA==xC||xB==xC,{Null,me},{Chop[tAbsorb],me}]]
]
```

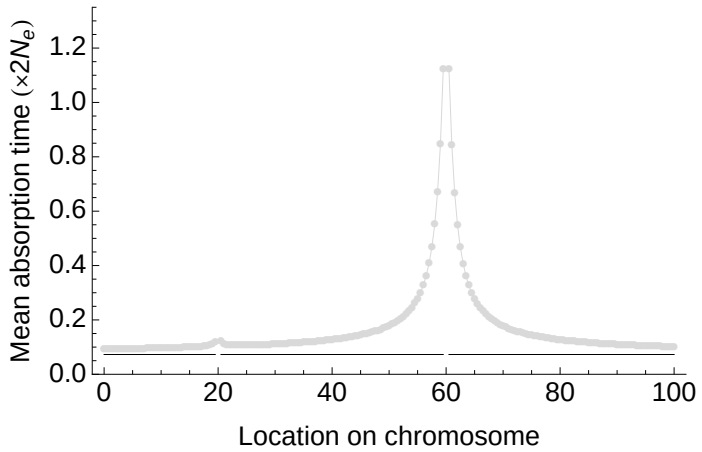
```
If[Max[Abs[Re[testEvalsFunc[mya, myb, mym, myNe, myxA, myxB, 25, sf]]]] < 0,
 -Max[Abs[Re[testEvalsFunc[mya, myb, mym, myNe, myxA, myxB, 25, sf]]]],
 Max[Abs[Re[testEvalsFunc[mya, myb, mym, myNe, myxA, myxB, 25, sf]]]]]
```

In[234]:=

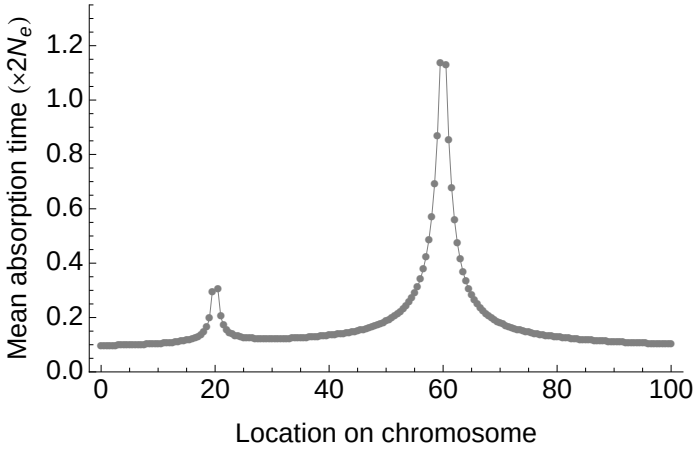
```
testEvalsFunc::usage="meanAbsTimeExactFunc[a, b, m, Ne, xA, xB, xC, scaleFac, ratio] r
testEvalsFunc[a_,b_,m_,Ne_,xA_,xB_,xC_,scaleFac_]:=Module[{evals,tAbsorb,rAC,rBC,x,JNA
rAC=Abs[xA-xC]scaleFac;
rBC=Abs[xB-xC]scaleFac;
JNACB={{-m,a,b,0},{1/(8 a (r1+r2)) m (-a^2+b^2+6 a (r1+r2)-4 m (r1+r2)-(r1+r2)^2+(a-b+r1+r2) √
JNABC={{-m,a,b,0},{m (-a^2+b^2+6 a r1-4 m r1-r1^2+(a-b+r1) √(-8 m r1+(a+b+r1)^2))/(8 a r1),a-m-r1-
JNCAB={{-m,b,a,0},{m (a^2-b^2+6 b r2-4 m r2-r2^2+(-a+b+r2) √(-8 m r2+(a+b+r2)^2))/(8 b r2),b-m-r1-r2
evals=If[xC<xA<xB,Eigenvalues[JNCAB],If[xA≤xC≤xB,Eigenvalues[JNACB],If[xA<xB<xC,Eigenv
Return[evals]
]
```

■ Various plots: $\mathcal{A}$ and $\mathcal{B}$ far apart

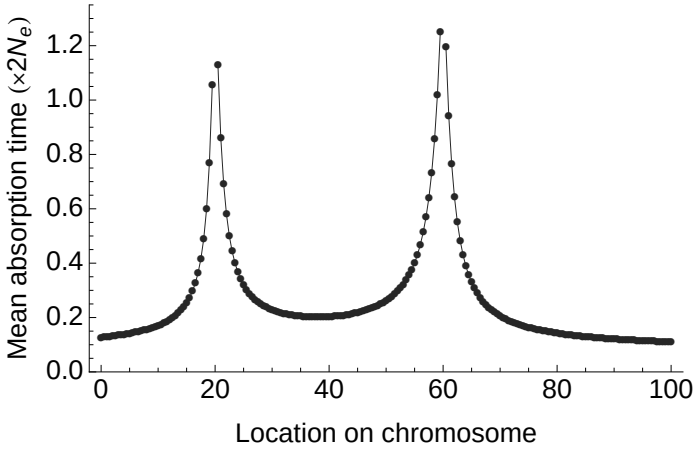
```
mya = 0.002;
myb = .4;
mym = 0.001;
myNe = 100;
myxA = 20;
myxB = 60;
myOffL = 20;
myOffR = 40;
sf = 0.01; (* r = 1 corresponds to 1/sf map units;
one map unit is equivalent to a recombination rate of sf. *)
dist = 0.5;
myRatio = 0.025;(*5;*)
yMax = 1.35;
plotCol = GrayLevel[0.85];
xVals = Range[myxA - myOffL, myxB + myOffR, dist];
Off[NIntegrate::zeroregion];
Off[NIntegrate::inumr];
Off[NIntegrate::slwcon];
Off[NIntegrate::ncvb];
Off[Power::infy];
Off[NIntegrate::inumri];
Off[Infinity::indet];
tBaseVals = meanAbsTimeFunc[0, 0, mym, myNe, myxA, myxB, #, sf, 0.0001] & /@ xVals;
tVals = meanAbsTimeFunc[mya, myb, mym, myNe, myxA, myxB, #, sf, myRatio] & /@ xVals;
tValsEx = meanAbsTimeExactFunc[mya, myb, mym, myNe, myxA, myxB, #, sf][[1]] & /@ xVals;
plotMeanAbsTime1Approx = ListPlot[{tVals},
    DataRange → {0, (myxB + myOffR + (myxA - myOffL))}, PlotRange → {Automatic, {0, yMax}},
    PlotMarkers → {{●, Tiny}, {""}}, PlotStyle → {plotCol}, Joined → False(*,Epilog→
     {{Dashed,Line[{{myxA,0},{myxA,10}}]},{Dashed,Line[{{myxB,0},{myxB,10}}]}}*),
    Frame → True, FrameStyle → {{Black, Opacity[0]}, {Black, Opacity[0]}},
    LabelStyle → {Directive[FontSize → 14], FontFamily → "Helvetica"},
    FrameLabel → {"Location on chromosome", "Mean absorption time (×2Ne)"}];
plotMeanAbsTime1Ex = ListPlot[{tValsEx, tBaseVals},
    DataRange → {0, (myxB + myOffR + (myxA - myOffL))}, PlotRange → {Automatic, {0, yMax}}
    (*,PlotMarkers→{{●,Tiny},{""}}*), PlotStyle → {plotCol, Black}, Joined → True,
    Frame → True, FrameStyle → {{Black, Opacity[0]}, {Black, Opacity[0]}}];
plotMeanAbsTime1 = Show[plotMeanAbsTime1Approx, plotMeanAbsTime1Ex]
On[NIntegrate::zeroregion];
On[NIntegrate::inumr];
On[NIntegrate::slwcon];
On[NIntegrate::ncvb];
On[Power::infy];
On[NIntegrate::inumri];
On[Infinity::indet];
```



```
mya = 0.02;
myb = .4;
mym = 0.001;
myNe = 100;
myxA = 20;
myxB = 60;
myOffL = 20;
myOffR = 40;
sf = 0.01; (* r = 1 corresponds to 1/sf map units;
one map unit is equivalent to a recombination rate of sf. *)
dist = 0.5;
myRatio = 0.025;(*5;*)
yMax = 1.35;
plotCol = GrayLevel[0.5];
xVals = Range[myxA - myOffL, myxB + myOffR, dist];
Off[NIntegrate::zeroregion];
Off[NIntegrate::inumr];
Off[NIntegrate::slwcon];
Off[NIntegrate::ncvb];
Off[Power::infy];
Off[NIntegrate::inumri];
Off[Infinity::indet];
tBaseVals = meanAbsTimeFunc[0, 0, mym, myNe, myxA, myxB, #, sf, 0.0001] & /@ xVals;
tVals = meanAbsTimeFunc[mya, myb, mym, myNe, myxA, myxB, #, sf, myRatio] & /@ xVals;
tValsEx = meanAbsTimeExactFunc[mya, myb, mym, myNe, myxA, myxB, #, sf][[1]] & /@ xVals;
plotMeanAbsTime2Approx = ListPlot[{tVals},
   DataRange → {0, (myxB + myOffR + (myxA - myOffL))}, PlotRange → {Automatic, {0, yMax}},
   PlotMarkers → {{●, Tiny}, {""}}, PlotStyle → {plotCol}, Joined → False(*,Epilog→
    {{Dashed,Line[{{myxA,0},{myxA,10}}]},{Dashed,Line[{{myxB,0},{myxB,10}}]}}*),
   Frame → True, FrameStyle → {{Black, Opacity[0]}, {Black, Opacity[0]}},
   LabelStyle → {Directive[FontSize → 14], FontFamily → "Helvetica"},
   FrameLabel → {"Location on chromosome", "Mean absorption time (×2N_e)"}];
plotMeanAbsTime2Ex = ListPlot[{tValsEx},
   DataRange → {0, (myxB + myOffR + (myxA - myOffL))}, PlotRange → {Automatic, {0, yMax}}
   (*,PlotMarkers→{{●,Tiny},{""}}*), PlotStyle → {plotCol}, Joined → True,
   Frame → True, FrameStyle → {{Black, Opacity[0]}, {Black, Opacity[0]}}];
plotMeanAbsTime2 = Show[plotMeanAbsTime2Approx, plotMeanAbsTime2Ex]
On[NIntegrate::zeroregion];
On[NIntegrate::inumr];
On[NIntegrate::slwcon];
On[NIntegrate::ncvb];
On[Power::infy];
On[NIntegrate::inumri];
On[Infinity::indet];
```



```
mya = 0.2;
myb = .4;
mym = 0.001;
myNe = 100;
myxA = 20;
myxB = 60;
myOffL = 20;
myOffR = 40;
sf = 0.01; (* r = 1 corresponds to 1/sf map units;
one map unit is equivalent to a recombination rate of sf. *)
dist = 0.5;
myRatio = 0.025;(*5;*)
yMax = 1.35;
plotCol = GrayLevel[0.15];
xVals = Range[myxA - myOffL, myxB + myOffR, dist];
Off[NIntegrate::zeroregion];
Off[NIntegrate::inumr];
Off[NIntegrate::slwcon];
Off[NIntegrate::ncvb];
Off[Power::infy];
Off[NIntegrate::inumri];
Off[Infinity::indet];
tBaseVals = meanAbsTimeFunc[0, 0, mym, myNe, myxA, myxB, #, sf, 0.0001] & /@ xVals;
tVals = meanAbsTimeFunc[mya, myb, mym, myNe, myxA, myxB, #, sf, myRatio] & /@ xVals;
tValsEx = meanAbsTimeExactFunc[mya, myb, mym, myNe, myxA, myxB, #, sf][[1]] & /@ xVals;
plotMeanAbsTime3Approx = ListPlot[{tVals},
   DataRange → {0, (myxB + myOffR + (myxA - myOffL))}, PlotRange → {Automatic, {0, yMax}},
   PlotMarkers → {{●, Tiny}, {""}}, PlotStyle → {plotCol}, Joined → False(*,Epilog→
    {{Dashed,Line[{{myxA,0},{myxA,10}}]},{Dashed,Line[{{myxB,0},{myxB,10}}]}}*),
   Frame → True, FrameStyle → {{Black, Opacity[0]}, {Black, Opacity[0]}},
   LabelStyle → {Directive[FontSize → 14], FontFamily → "Helvetica"},
   FrameLabel → {"Location on chromosome", "Mean absorption time (×2N_e)"}];
plotMeanAbsTime3Ex = ListPlot[{tValsEx},
   DataRange → {0, (myxB + myOffR + (myxA - myOffL))}, PlotRange → {Automatic, {0, yMax}}
   (*,PlotMarkers→{{●,Tiny},{""}}*), PlotStyle → {plotCol}, Joined → True,
   Frame → True, FrameStyle → {{Black, Opacity[0]}, {Black, Opacity[0]}}];
plotMeanAbsTime3 = Show[plotMeanAbsTime3Approx, plotMeanAbsTime3Ex]
On[NIntegrate::zeroregion];
On[NIntegrate::inumr];
On[NIntegrate::slwcon];
On[NIntegrate::ncvb];
On[Power::infy];
On[NIntegrate::inumri];
On[Infinity::indet];
```



```
plotMeanAbsTimeA = Labeled[Show[plotMeanAbsTime1, plotMeanAbsTime2, plotMeanAbsTime3],
  "A", {{Top, Left}}, FrameMargins → {{14, 0}, {0, 0}},
  LabelStyle → {Directive[FontSize → 18, Bold], FontFamily → "Helvetica"}]
```

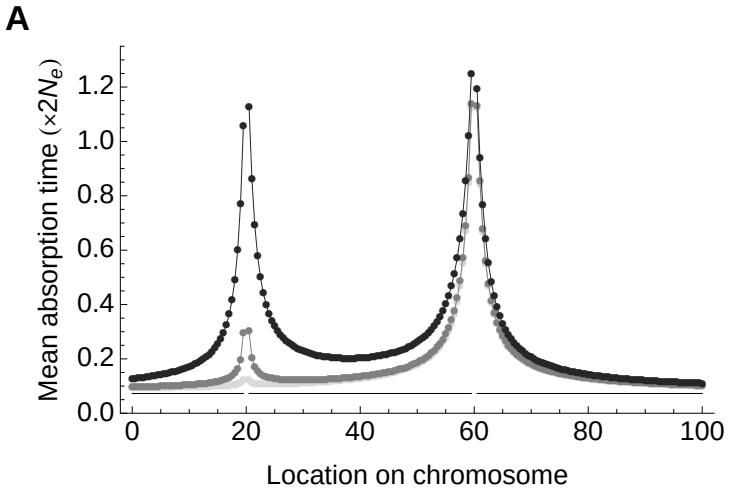


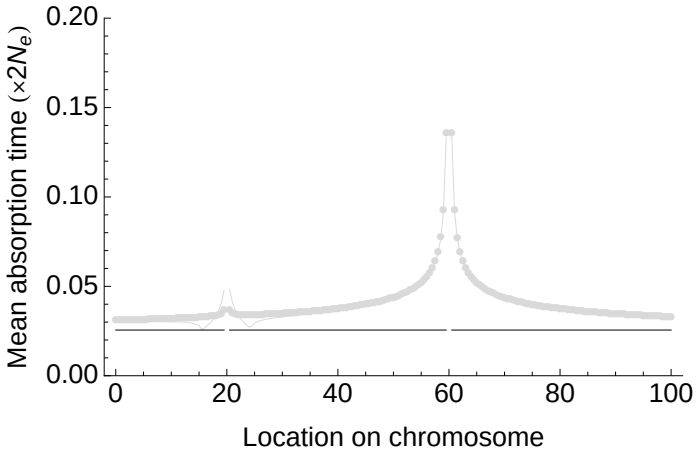
Figure 1: The effect of linkage to two selected sites on the mean absorption time of a neutral de-novo mutation. The loci $\mathcal{A}$ and $\mathcal{B}$ under selection are located 20 and 60 map units from the left end of the chromosome, respectively. It is arbitrarily assumed that one map unit (cM) corresponds to a recombination rate of $r = 0.01$ and the effective size of the island population is $N_e = 100$. The scaled selection coefficient in favour of B_1 is $\beta = 2 N_e b = 80$ and the scaled migration rate is $\mu = 2 N_e m = 0.2$. From light to dark, α / β takes values of 0.005, 0.05 and 0.5, where $\alpha = 2 N_e a$ is the scaled selection coefficient in favour of A_1. Points show values computed using the approximate effective migration rates in equations (1) to (3) and lines are based on numerically computed exact effective migration rates. The horizontal black line denotes the baseline for no linkage to selected sites.

```
Export[PlotPath <> "meanAbsTime_neutrSite_A.eps", plotMeanAbsTimeA, "EPS"]
```

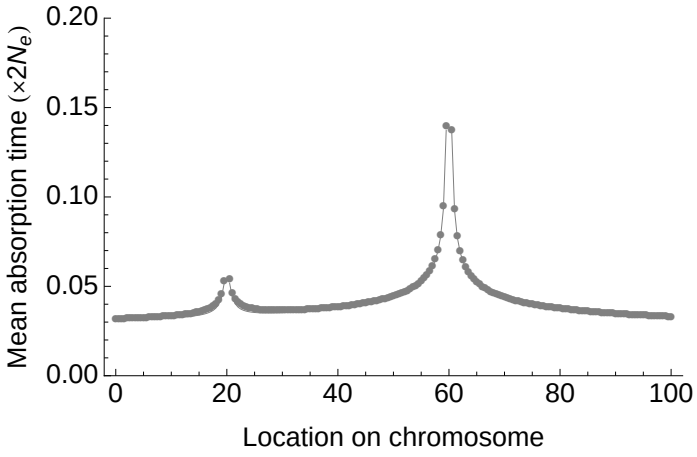
```
/Users/Simon/Documents/LocAdD/results/130419/linkedNeutrSite/figures/
  meanAbsTime_neutrSite_A.eps
```

Remark: It is arbitrarily assumed here that one map unit corresponds to a recombination rate of $r = 0.01$. With $N_e = 100$, this results in $\rho = 2 N_e r = 2$. We are in a continuous-time model here and r is the recombination *rate*, but for a rough comparison, we may assume that r represents the probability of a recombination event per generation, such that the definition of a centimorgan (cM) [equal to one map unit] as being equivalent to one recombination events in a hundred generations ($r = 0.01$) applies. Assuming that one map unit corresponds to about one mega base pair (Mb), which comes close to estimates for humans (Kulathinal *et al.* 2008) or *Drosophila pseudoobscura* (Scott *et al.* 2004), the hypothetical chromosome depicted in the figure above would consist of about 100 Mb. This is in the range of human chromosomes.

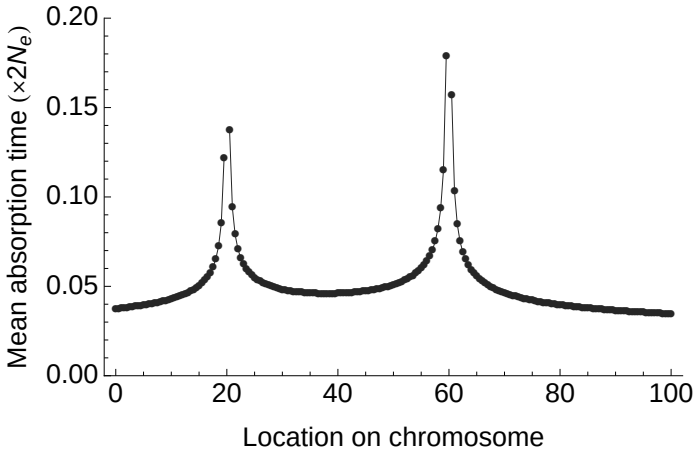
```
mya = 0.002;
myb = .4;
mym = 0.024;
myNe = 100;
myxA = 20;
myxB = 60;
myOffL = 20;
myOffR = 40;
sf = 0.01; (* r = 1 corresponds to 1/sf map units;
one map unit is equivalent to a recombination rate of sf. *)
dist = 0.5;
myRatio = 0.025;(*5;*)
yMax = 0.2;
plotCol = GrayLevel[0.85];
xVals = Range[myxA - myOffL, myxB + myOffR, dist];
Off[NIntegrate::zeroregion];
Off[NIntegrate::inumr];
Off[NIntegrate::slwcon];
Off[NIntegrate::ncvb];
Off[Power::infy];
Off[NIntegrate::inumri];
Off[Infinity::indet];
tBaseVals = meanAbsTimeFunc[0, 0, mym, myNe, myxA, myxB, #, sf, 0.0001] & /@ xVals;
tVals = meanAbsTimeFunc[mya, myb, mym, myNe, myxA, myxB, #, sf, myRatio] & /@ xVals;
tValsEx = meanAbsTimeExactFunc[mya, myb, mym, myNe, myxA, myxB, #, sf][[1]] & /@ xVals;
plotMeanAbsTime4Approx = ListPlot[{tVals},
   DataRange → {0, (myxB + myOffR + (myxA - myOffL))}, PlotRange → {Automatic, {0, yMax}},
   PlotMarkers → {{●, Tiny}, {""}}, PlotStyle → {plotCol}, Joined → False(*,Epilog→
    {{Dashed,Line[{{myxA,0},{myxA,10}}]},{Dashed,Line[{{myxB,0},{myxB,10}}]}}*),
   Frame → True, FrameStyle → {{Black, Opacity[0]}, {Black, Opacity[0]}},
   LabelStyle → {Directive[FontSize → 14], FontFamily → "Helvetica"},
   FrameLabel → {"Location on chromosome", "Mean absorption time (×2N_e)"}];
plotMeanAbsTime4Ex = ListPlot[{tValsEx, tBaseVals},
   DataRange → {0, (myxB + myOffR + (myxA - myOffL))}, PlotRange → {Automatic, {0, yMax}}
   (*,PlotMarkers→{{●,Tiny},{""}}*), PlotStyle → {plotCol, Black}, Joined → True,
   Frame → True, FrameStyle → {{Black, Opacity[0]}, {Black, Opacity[0]}}];
plotMeanAbsTime4 = Show[plotMeanAbsTime4Approx, plotMeanAbsTime4Ex]
On[NIntegrate::zeroregion];
On[NIntegrate::inumr];
On[NIntegrate::slwcon];
On[NIntegrate::ncvb];
On[Power::infy];
On[NIntegrate::inumri];
On[Infinity::indet];
```



```
mya = 0.02;
myb = .4;
mym = 0.024;
myNe = 100;
myxA = 20;
myxB = 60;
myOffL = 20;
myOffR = 40;
sf = 0.01; (* r = 1 corresponds to 1/sf map units;
one map unit is equivalent to a recombination rate of sf. *)
dist = 0.5;
myRatio = 0.025;(*5;*)
yMax = 0.2;
plotCol = GrayLevel[0.5];
xVals = Range[myxA - myOffL, myxB + myOffR, dist];
Off[NIntegrate::zeroregion];
Off[NIntegrate::inumr];
Off[NIntegrate::slwcon];
Off[NIntegrate::ncvb];
Off[Power::infy];
Off[NIntegrate::inumri];
Off[Infinity::indet];
tBaseVals = meanAbsTimeFunc[0, 0, mym, myNe, myxA, myxB, #, sf, 0.0001] & /@ xVals;
tVals = meanAbsTimeFunc[mya, myb, mym, myNe, myxA, myxB, #, sf, myRatio] & /@ xVals;
tValsEx = meanAbsTimeExactFunc[mya, myb, mym, myNe, myxA, myxB, #, sf][[1]] & /@ xVals;
plotMeanAbsTime5Approx = ListPlot[{tVals},
   DataRange → {0, (myxB + myOffR + (myxA - myOffL))}, PlotRange → {Automatic, {0, yMax}},
   PlotMarkers → {{●, Tiny}, {""}}, PlotStyle → {plotCol}, Joined → False(*,Epilog→
    {{Dashed,Line[{{myxA,0},{myxA,10}}]},{Dashed,Line[{{myxB,0},{myxB,10}}]}}*),
   Frame → True, FrameStyle → {{Black, Opacity[0]}, {Black, Opacity[0]}},
   LabelStyle → {Directive[FontSize → 14], FontFamily → "Helvetica"},
   FrameLabel → {"Location on chromosome", "Mean absorption time (×2N_e)"}];
plotMeanAbsTime5Ex = ListPlot[{tValsEx},
   DataRange → {0, (myxB + myOffR + (myxA - myOffL))}, PlotRange → {Automatic, {0, yMax}}
   (*,PlotMarkers→{{●,Tiny},{""}}*), PlotStyle → {plotCol}, Joined → True,
   Frame → True, FrameStyle → {{Black, Opacity[0]}, {Black, Opacity[0]}}];
plotMeanAbsTime5 = Show[plotMeanAbsTime5Approx, plotMeanAbsTime5Ex]
On[NIntegrate::zeroregion];
On[NIntegrate::inumr];
On[NIntegrate::slwcon];
On[NIntegrate::ncvb];
On[Power::infy];
On[NIntegrate::inumri];
On[Infinity::indet];
```



```
mya = 0.2;
myb = .4;
mym = 0.024;
myNe = 100;
myxA = 20;
myxB = 60;
myOffL = 20;
myOffR = 40;
sf = 0.01; (* r = 1 corresponds to 1/sf map units;
one map unit is equivalent to a recombination rate of sf. *)
dist = 0.5;
myRatio = 0.025;(*5;*)
yMax = 0.2;
plotCol = GrayLevel[0.15];
xVals = Range[myxA - myOffL, myxB + myOffR, dist];
Off[NIntegrate::zeroregion];
Off[NIntegrate::inumr];
Off[NIntegrate::slwcon];
Off[NIntegrate::ncvb];
Off[Power::infy];
Off[NIntegrate::inumri];
Off[Infinity::indet];
tBaseVals = meanAbsTimeFunc[0, 0, mym, myNe, myxA, myxB, #, sf, 0.0001] & /@ xVals;
tVals = meanAbsTimeFunc[mya, myb, mym, myNe, myxA, myxB, #, sf, myRatio] & /@ xVals;
tValsEx = meanAbsTimeExactFunc[mya, myb, mym, myNe, myxA, myxB, #, sf][[1]] & /@ xVals;
plotMeanAbsTime6Approx = ListPlot[{tVals},
   DataRange → {0, (myxB + myOffR + (myxA - myOffL))}, PlotRange → {Automatic, {0, yMax}},
   PlotMarkers → {{●, Tiny}, {""}}, PlotStyle → {plotCol}, Joined → False(*,Epilog→
    {{Dashed,Line[{{myxA,0},{myxA,10}}]},{Dashed,Line[{{myxB,0},{myxB,10}}]}}*),
   Frame → True, FrameStyle → {{Black, Opacity[0]}, {Black, Opacity[0]}},
   LabelStyle → {Directive[FontSize → 14], FontFamily → "Helvetica"},
   FrameLabel → {"Location on chromosome", "Mean absorption time (×2N_e)"}];
plotMeanAbsTime6Ex = ListPlot[{tValsEx},
   DataRange → {0, (myxB + myOffR + (myxA - myOffL))}, PlotRange → {Automatic, {0, yMax}}
   (*,PlotMarkers→{{●,Tiny},{""}}*), PlotStyle → {plotCol}, Joined → True,
   Frame → True, FrameStyle → {{Black, Opacity[0]}, {Black, Opacity[0]}}];
plotMeanAbsTime6 = Show[plotMeanAbsTime6Approx, plotMeanAbsTime6Ex]
On[NIntegrate::zeroregion];
On[NIntegrate::inumr];
On[NIntegrate::slwcon];
On[NIntegrate::ncvb];
On[Power::infy];
On[NIntegrate::inumri];
On[Infinity::indet];
```



```
plotMeanAbsTimeB = Labeled[Show[plotMeanAbsTime4, plotMeanAbsTime5, plotMeanAbsTime6],
  "B", {{Top, Left}}, FrameMargins → {{14, 0}, {0, 0}},
  LabelStyle → {Directive[FontSize → 18, Bold], FontFamily → "Helvetica"}]
```

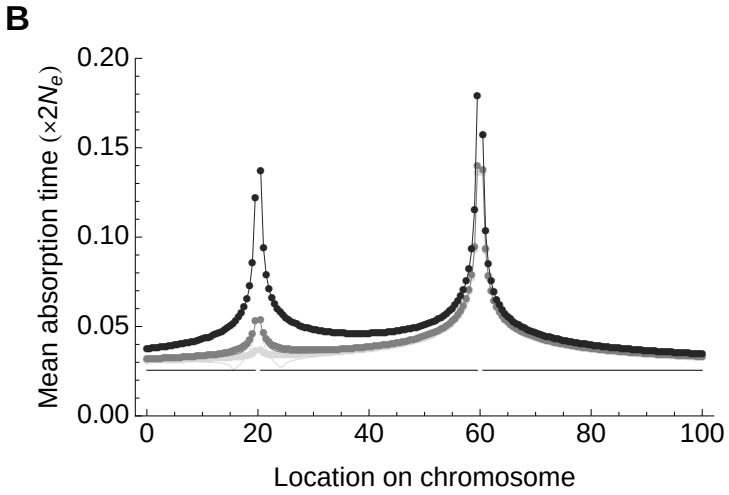
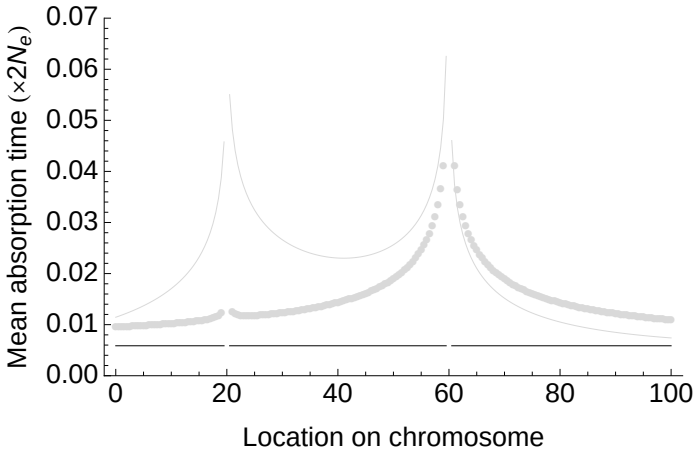


Figure 2: The effect of linkage to two selected sites on the mean absorption time of a neutral de-novo mutation. The loci $\mathcal{A}$ and $\mathcal{B}$ under selection are located 20 and 60 map units from the left end of the chromosome, respectively. It is arbitrarily assumed that one map unit (cM) corresponds to a recombination rate of $r = 0.01$ and the effective size of the island population is $N_e = 100$. The scaled selection coefficient in favour of B_1 is $\beta = 2 N_e b = 80$ and the scaled migration rate is $\mu = 2 N_e m = 4.8$. From light to dark, α / β takes values of 0.005, 0.05 and 0.5, where $\alpha = 2 N_e a$ is the scaled selection coefficient in favour of A_1. Points show values computed using the approximate effective migration rates in equations (1) to (3) and lines are based on numerically computed exact effective migration rates. The horizontal black line denotes the baseline for no linkage to selected sites.

```
Export[PlotPath <> "meanAbsTime_neutrSite_B.eps", plotMeanAbsTimeB, "EPS"]
```

```
/Users/Simon/Documents/LocAdD/results/130419/linkedNeutrSite/figures/
  meanAbsTime_neutrSite_B.eps
```

```
mya = 0.002;
myb = .4;
mym = 0.24;
myNe = 100;
myxA = 20;
myxB = 60;
myOffL = 20;
myOffR = 40;
sf = 0.01; (* r = 1 corresponds to 1/sf map units;
one map unit is equivalent to a recombination rate of sf. *)
dist = 0.5;
myRatio = 0.025;(*5;*)
yMax = 0.07;
plotCol = GrayLevel[0.85];
xVals = Range[myxA - myOffL, myxB + myOffR, dist];
Off[NIntegrate::zeroregion];
Off[NIntegrate::inumr];
Off[NIntegrate::slwcon];
Off[NIntegrate::ncvb];
Off[Power::infy];
Off[NIntegrate::inumri];
Off[Infinity::indet];
tBaseVals = meanAbsTimeFunc[0, 0, mym, myNe, myxA, myxB, #, sf, 0.0001] & /@ xVals;
tVals = meanAbsTimeFunc[mya, myb, mym, myNe, myxA, myxB, #, sf, myRatio] & /@ xVals;
tValsEx = meanAbsTimeExactFunc[mya, myb, mym, myNe, myxA, myxB, #, sf][[1]] & /@ xVals;
plotMeanAbsTime7Approx = ListPlot[{tVals},
    DataRange → {0, (myxB + myOffR + (myxA - myOffL))}, PlotRange → {Automatic, {0, yMax}},
    PlotMarkers → {{●, Tiny}, {""}}, PlotStyle → {plotCol}, Joined → False(*,Epilog→
     {{Dashed,Line[{{myxA,0},{myxA,10}}]},{Dashed,Line[{{myxB,0},{myxB,10}}]}}*),
    Frame → True, FrameStyle → {{Black, Opacity[0]}, {Black, Opacity[0]}},
    LabelStyle → {Directive[FontSize → 14], FontFamily → "Helvetica"},
    FrameLabel → {"Location on chromosome", "Mean absorption time (×2N_e)"}];
plotMeanAbsTime7Ex = ListPlot[{tValsEx, tBaseVals},
    DataRange → {0, (myxB + myOffR + (myxA - myOffL))}, PlotRange → {Automatic, {0, yMax}}
    (*,PlotMarkers→{{●,Tiny},{""}}*), PlotStyle → {plotCol, Black}, Joined → True,
    Frame → True, FrameStyle → {{Black, Opacity[0]}, {Black, Opacity[0]}}];
plotMeanAbsTime7 = Show[plotMeanAbsTime7Approx, plotMeanAbsTime7Ex]
On[NIntegrate::zeroregion];
On[NIntegrate::inumr];
On[NIntegrate::slwcon];
On[NIntegrate::ncvb];
On[Power::infy];
On[NIntegrate::inumri];
On[Infinity::indet];
```



Remark: It is arbitrarily assumed here that one map unit corresponds to a recombination rate of $r = 0.01$.

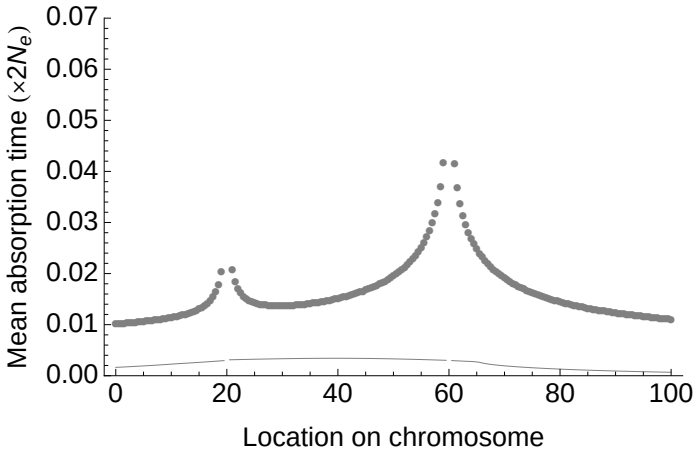
```
testEvalsFunc[mya, myb, mym, myNe, myxA, myxB, 25, sf] // TableForm
-Max[Re[testEvalsFunc[mya, myb, mym, myNe, myxA, myxB, 25, sf]]]
If[Max[Abs[Re[testEvalsFunc[mya, myb, mym, myNe, myxA, myxB, 25, sf]]]] < 0,
 -Max[Abs[Re[testEvalsFunc[mya, myb, mym, myNe, myxA, myxB, 25, sf]]]],
 Max[Abs[Re[testEvalsFunc[mya, myb, mym, myNe, myxA, myxB, 25, sf]]]]]
```

```
-0.388589 - 0.243451 i
-0.388438 - 0.0971767 i
-0.012562 - 0.0794557 i
-0.012411 + 0.0668189 i
```

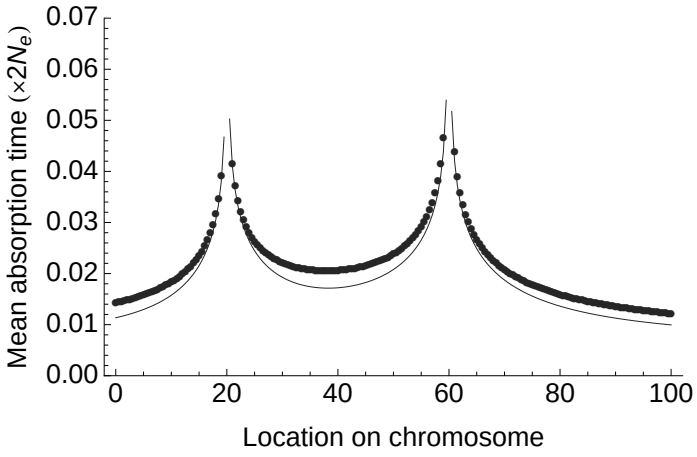
```
0.012411
```

```
0.388589
```

```
mya = 0.02;
myb = .4;
mym = 0.24;
myNe = 100;
myxA = 20;
myxB = 60;
myOffL = 20;
myOffR = 40;
sf = 0.01; (* r = 1 corresponds to 1/sf map units;
one map unit is equivalent to a recombination rate of sf. *)
dist = 0.5;
myRatio = 0.025;(*5;*)
yMax = 0.07;
plotCol = GrayLevel[0.5];
xVals = Range[myxA - myOffL, myxB + myOffR, dist];
Off[NIntegrate::zeroregion];
Off[NIntegrate::inumr];
Off[NIntegrate::slwcon];
Off[NIntegrate::ncvb];
Off[Power::infy];
Off[NIntegrate::inumri];
Off[Infinity::indet];
tBaseVals = meanAbsTimeFunc[0, 0, mym, myNe, myxA, myxB, #, sf, 0.0001] & /@ xVals;
tVals = meanAbsTimeFunc[mya, myb, mym, myNe, myxA, myxB, #, sf, myRatio] & /@ xVals;
tValsEx = meanAbsTimeExactFunc[mya, myb, mym, myNe, myxA, myxB, #, sf][[1]] & /@ xVals;
plotMeanAbsTime8Approx = ListPlot[{tVals},
    DataRange → {0, (myxB + myOffR + (myxA - myOffL))}, PlotRange → {Automatic, {0, yMax}},
    PlotMarkers → {{●, Tiny}, {""}}, PlotStyle → {plotCol}, Joined → False(*,Epilog→
     {{Dashed,Line[{{myxA,0},{myxA,10}}]},{Dashed,Line[{{myxB,0},{myxB,10}}]}}*),
    Frame → True, FrameStyle → {{Black, Opacity[0]}, {Black, Opacity[0]}},
    LabelStyle → {Directive[FontSize → 14], FontFamily → "Helvetica"},
    FrameLabel → {"Location on chromosome", "Mean absorption time (×2N_e)"}];
plotMeanAbsTime8Ex = ListPlot[{tValsEx},
    DataRange → {0, (myxB + myOffR + (myxA - myOffL))}, PlotRange → {Automatic, {0, yMax}}
    (*,PlotMarkers→{{●,Tiny},{""}}*), PlotStyle → {plotCol}, Joined → True,
    Frame → True, FrameStyle → {{Black, Opacity[0]}, {Black, Opacity[0]}}];
plotMeanAbsTime8 = Show[plotMeanAbsTime8Approx, plotMeanAbsTime8Ex]
On[NIntegrate::zeroregion];
On[NIntegrate::inumr];
On[NIntegrate::slwcon];
On[NIntegrate::ncvb];
On[Power::infy];
On[NIntegrate::inumri];
On[Infinity::indet];
```



```
mya = 0.2;
myb = .4;
mym = 0.24;
myNe = 100;
myxA = 20;
myxB = 60;
myOffL = 20;
myOffR = 40;
sf = 0.01; (* r = 1 corresponds to 1/sf map units;
one map unit is equivalent to a recombination rate of sf. *)
dist = 0.5;
myRatio = 0.025;(*5;*)
yMax = 0.07;
plotCol = GrayLevel[0.15];
xVals = Range[myxA - myOffL, myxB + myOffR, dist];
Off[NIntegrate::zeroregion];
Off[NIntegrate::inumr];
Off[NIntegrate::slwcon];
Off[NIntegrate::ncvb];
Off[Power::infy];
Off[NIntegrate::inumri];
Off[Infinity::indet];
tBaseVals = meanAbsTimeFunc[0, 0, mym, myNe, myxA, myxB, #, sf, 0.0001] & /@ xVals;
tVals = meanAbsTimeFunc[mya, myb, mym, myNe, myxA, myxB, #, sf, myRatio] & /@ xVals;
tValsEx = meanAbsTimeExactFunc[mya, myb, mym, myNe, myxA, myxB, #, sf][[1]] & /@ xVals;
plotMeanAbsTime9Approx = ListPlot[{tVals},
   DataRange → {0, (myxB + myOffR + (myxA - myOffL))}, PlotRange → {Automatic, {0, yMax}},
   PlotMarkers → {{●, Tiny}, {""}}, PlotStyle → {plotCol}, Joined → False(*,Epilog→
    {{Dashed,Line[{{myxA,0},{myxA,10}}]},{Dashed,Line[{{myxB,0},{myxB,10}}]}}*),
   Frame → True, FrameStyle → {{Black, Opacity[0]}, {Black, Opacity[0]}},
   LabelStyle → {Directive[FontSize → 14], FontFamily → "Helvetica"},
   FrameLabel → {"Location on chromosome", "Mean absorption time (×2Ne)"}];
plotMeanAbsTime9Ex = ListPlot[{tValsEx},
   DataRange → {0, (myxB + myOffR + (myxA - myOffL))}, PlotRange → {Automatic, {0, yMax}}
   (*,PlotMarkers→{{●,Tiny},{""}}*), PlotStyle → {plotCol}, Joined → True,
   Frame → True, FrameStyle → {{Black, Opacity[0]}, {Black, Opacity[0]}}];
plotMeanAbsTime9 = Show[plotMeanAbsTime9Approx, plotMeanAbsTime9Ex]
On[NIntegrate::zeroregion];
On[NIntegrate::inumr];
On[NIntegrate::slwcon];
On[NIntegrate::ncvb];
On[Power::infy];
On[NIntegrate::inumri];
On[Infinity::indet];
```



Remark: It is arbitrarily assumed here that one map unit corresponds to a recombination rate of $r = 0.01$.

```
plotMeanAbsTimeC = Labeled[Show[plotMeanAbsTime7, plotMeanAbsTime8, plotMeanAbsTime9],
  "C", {{Top, Left}}, FrameMargins → {{14, 0}, {0, 0}},
  LabelStyle → {Directive[FontSize → 18, Bold], FontFamily → "Helvetica"}]
```

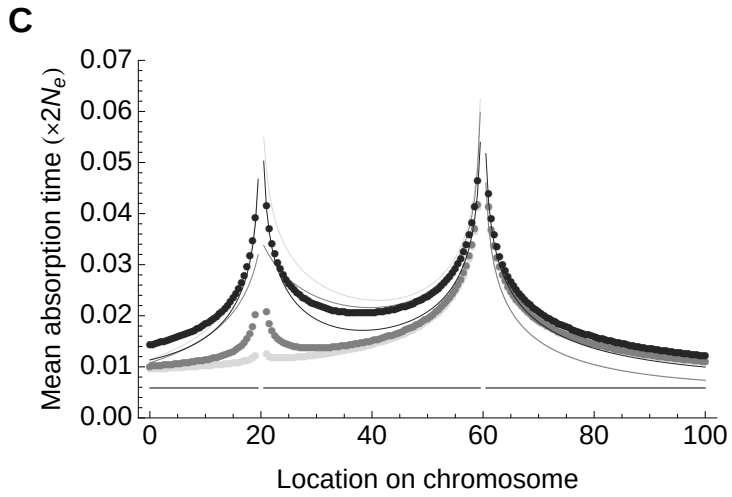


Figure 3: The effect of linkage to two selected sites on the mean absorption time of a neutral de-novo mutation. The biallelic loci $\mathcal{A}$ and $\mathcal{B}$ under selection are located 40 and 60 map units from the left end of the chromosome, respectively. It is arbitrarily assumed that one map unit corresponds to a recombination rate of $r = 0.01$ and the effective size of the island population is $N_e = 100$. The scaled selection coefficient in favour of B_1 is $\beta = 2 N_e b = 80$ and the scaled migration rate is $\mu = 2 N_e m = 48$. From light to dark, α / β takes values of 0.005, 0.05 and 0.5, where $\alpha = 2 N_e a$ is the scaled selection coefficient in favour of A_1. The effective migration rate was computed according to equations (1) to (3). Points are shown only if the assumption of weak migration is justified ($m / r \leq 0.2$), which is not the case if the neutral locus is very close to one of the selected sites. The horizontal black line denotes the baseline for no linkage to selected sites.

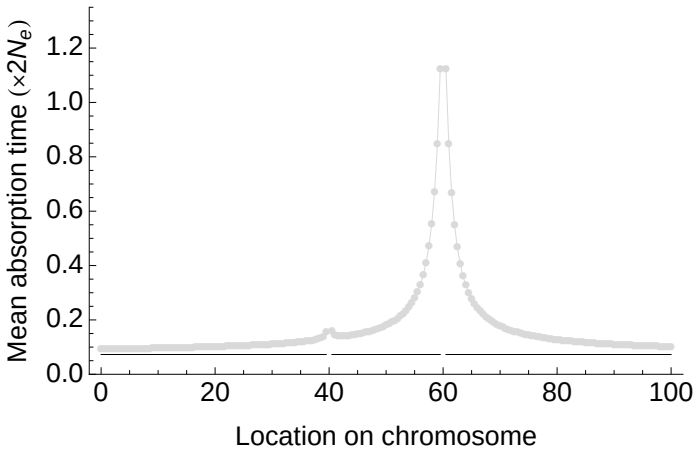
Remark: It is arbitrarily assumed here that one map unit corresponds to a recombination rate of $r = 0.01$.

```
Export[PlotPath <> "meanAbsTime_neutrSite_C.eps", plotMeanAbsTimeC, "EPS"]
```

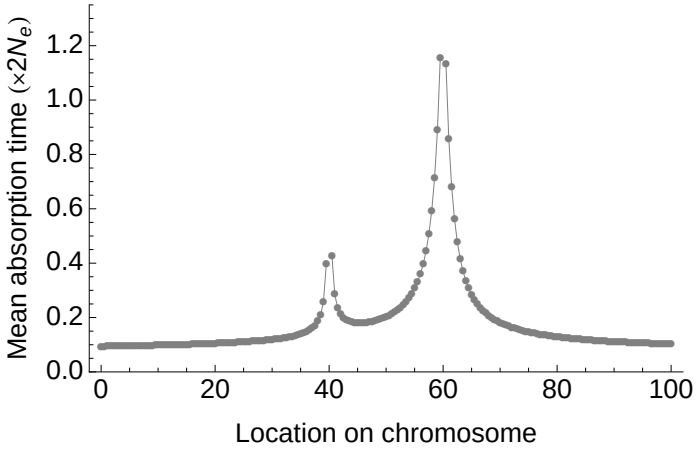
```
/Users/Simon/Documents/LocAdD/results/130419/linkedNeutrSite/figures/
  meanAbsTime_neutrSite_C.eps
```

■ Various plots: $\mathcal{A}$ and $\mathcal{B}$ close

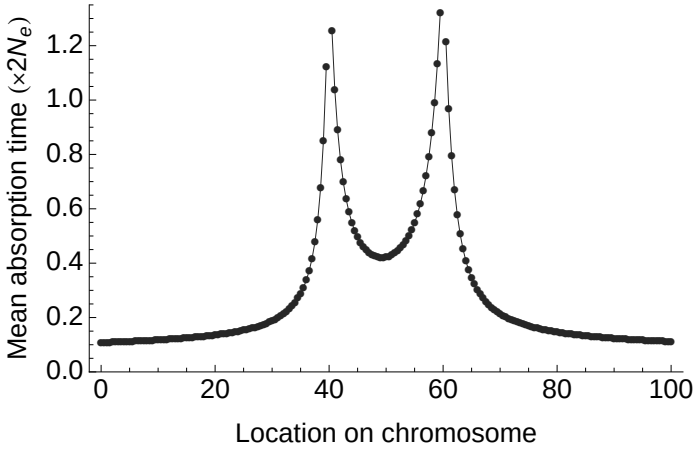
```
mya = 0.002;
myb = .4;
mym = 0.001;
myNe = 100;
myxA = 40;
myxB = 60;
myOffL = 40;
myOffR = 40;
sf = 0.01; (* r = 1 corresponds to 1/sf map units;
one map unit is equivalent to a recombination rate of sf. *)
dist = 0.5;
myRatio = 0.025;(*5;*)
yMax = 1.35;
plotCol = GrayLevel[0.85];
xVals = Range[myxA - myOffL, myxB + myOffR, dist];
Off[NIntegrate::zeroregion];
Off[NIntegrate::inumr];
Off[NIntegrate::slwcon];
Off[NIntegrate::ncvb];
Off[Power::infy];
Off[NIntegrate::inumri];
Off[Infinity::indet];
tBaseVals = meanAbsTimeFunc[0, 0, mym, myNe, myxA, myxB, #, sf, 0.0001] & /@ xVals;
tVals = meanAbsTimeFunc[mya, myb, mym, myNe, myxA, myxB, #, sf, myRatio] & /@ xVals;
tValsEx = meanAbsTimeExactFunc[mya, myb, mym, myNe, myxA, myxB, #, sf][[1]] & /@ xVals;
plotMeanAbsTime10Approx = ListPlot[{tVals},
    DataRange → {0, (myxB + myOffR + (myxA - myOffL))}, PlotRange → {Automatic, {0, yMax}},
    PlotMarkers → {{●, Tiny}, {""}}, PlotStyle → {plotCol}, Joined → False(*,Epilog→
     {{Dashed,Line[{{myxA,0},{myxA,10}}]},{Dashed,Line[{{myxB,0},{myxB,10}}]}}*),
    Frame → True, FrameStyle → {{Black, Opacity[0]}, {Black, Opacity[0]}},
    LabelStyle → {Directive[FontSize → 14], FontFamily → "Helvetica"},
    FrameLabel → {"Location on chromosome", "Mean absorption time (×2Ne)"}];
plotMeanAbsTime10Ex = ListPlot[{tValsEx, tBaseVals},
    DataRange → {0, (myxB + myOffR + (myxA - myOffL))}, PlotRange → {Automatic, {0, yMax}}
    (*,PlotMarkers→{{●,Tiny},{""}}*), PlotStyle → {plotCol, Black}, Joined → True,
    Frame → True, FrameStyle → {{Black, Opacity[0]}, {Black, Opacity[0]}}];
plotMeanAbsTime10 = Show[plotMeanAbsTime10Approx, plotMeanAbsTime10Ex]
On[NIntegrate::zeroregion];
On[NIntegrate::inumr];
On[NIntegrate::slwcon];
On[NIntegrate::ncvb];
On[Power::infy];
On[NIntegrate::inumri];
On[Infinity::indet];
```



```
mya = 0.02;
myb = .4;
mym = 0.001;
myNe = 100;
myxA = 40;
myxB = 60;
myOffL = 40;
myOffR = 40;
sf = 0.01; (* r = 1 corresponds to 1/sf map units;
one map unit is equivalent to a recombination rate of sf. *)
dist = 0.5;
myRatio = 0.025;(*5;*)
yMax = 1.35;
plotCol = GrayLevel[0.5];
xVals = Range[myxA - myOffL, myxB + myOffR, dist];
Off[NIntegrate::zeroregion];
Off[NIntegrate::inumr];
Off[NIntegrate::slwcon];
Off[NIntegrate::ncvb];
Off[Power::infy];
Off[NIntegrate::inumri];
Off[Infinity::indet];
tBaseVals = meanAbsTimeFunc[0, 0, mym, myNe, myxA, myxB, #, sf, 0.0001] & /@ xVals;
tVals = meanAbsTimeFunc[mya, myb, mym, myNe, myxA, myxB, #, sf, myRatio] & /@ xVals;
tValsEx = meanAbsTimeExactFunc[mya, myb, mym, myNe, myxA, myxB, #, sf][[1]] & /@ xVals;
plotMeanAbsTime11Approx = ListPlot[{tVals},
    DataRange → {0, (myxB + myOffR + (myxA - myOffL))}, PlotRange → {Automatic, {0, yMax}},
    PlotMarkers → {{●, Tiny}, {""}}, PlotStyle → {plotCol}, Joined → False(*,Epilog→
     {{Dashed,Line[{{myxA,0},{myxA,10}}]},{Dashed,Line[{{myxB,0},{myxB,10}}]}}*),
    Frame → True, FrameStyle → {{Black, Opacity[0]}, {Black, Opacity[0]}},
    LabelStyle → {Directive[FontSize → 14], FontFamily → "Helvetica"},
    FrameLabel → {"Location on chromosome", "Mean absorption time (×2N_e)"}];
plotMeanAbsTime11Ex = ListPlot[{tValsEx},
    DataRange → {0, (myxB + myOffR + (myxA - myOffL))}, PlotRange → {Automatic, {0, yMax}}
    (*,PlotMarkers→{{●,Tiny},{""}}*), PlotStyle → {plotCol}, Joined → True,
    Frame → True, FrameStyle → {{Black, Opacity[0]}, {Black, Opacity[0]}}];
plotMeanAbsTime11 = Show[plotMeanAbsTime11Approx, plotMeanAbsTime11Ex]
On[NIntegrate::zeroregion];
On[NIntegrate::inumr];
On[NIntegrate::slwcon];
On[NIntegrate::ncvb];
On[Power::infy];
On[NIntegrate::inumri];
On[Infinity::indet];
```



```
mya = 0.2;
myb = .4;
mym = 0.001;
myNe = 100;
myxA = 40;
myxB = 60;
myOffL = 40;
myOffR = 40;
sf = 0.01; (* r = 1 corresponds to 1/sf map units;
one map unit is equivalent to a recombination rate of sf. *)
dist = 0.5;
myRatio = 0.025;(*5;*)
yMax = 1.35;
plotCol = GrayLevel[0.15];
xVals = Range[myxA - myOffL, myxB + myOffR, dist];
Off[NIntegrate::zeroregion];
Off[NIntegrate::inumr];
Off[NIntegrate::slwcon];
Off[NIntegrate::ncvb];
Off[Power::infy];
Off[NIntegrate::inumri];
Off[Infinity::indet];
tBaseVals = meanAbsTimeFunc[0, 0, mym, myNe, myxA, myxB, #, sf, 0.0001] & /@ xVals;
tVals = meanAbsTimeFunc[mya, myb, mym, myNe, myxA, myxB, #, sf, myRatio] & /@ xVals;
tValsEx = meanAbsTimeExactFunc[mya, myb, mym, myNe, myxA, myxB, #, sf][[1]] & /@ xVals;
plotMeanAbsTime12Approx = ListPlot[{tVals},
    DataRange → {0, (myxB + myOffR + (myxA - myOffL))}, PlotRange → {Automatic, {0, yMax}},
    PlotMarkers → {{●, Tiny}, {""}}, PlotStyle → {plotCol}, Joined → False(*,Epilog→
     {{Dashed,Line[{{myxA,0},{myxA,10}}]},{Dashed,Line[{{myxB,0},{myxB,10}}]}}*),
    Frame → True, FrameStyle → {{Black, Opacity[0]}, {Black, Opacity[0]}},
    LabelStyle → {Directive[FontSize → 14], FontFamily → "Helvetica"},
    FrameLabel → {"Location on chromosome", "Mean absorption time (×2N_e)"}];
plotMeanAbsTime12Ex = ListPlot[{tValsEx},
    DataRange → {0, (myxB + myOffR + (myxA - myOffL))}, PlotRange → {Automatic, {0, yMax}}
    (*,PlotMarkers→{{●,Tiny},{""}}*), PlotStyle → {plotCol}, Joined → True,
    Frame → True, FrameStyle → {{Black, Opacity[0]}, {Black, Opacity[0]}}];
plotMeanAbsTime12 = Show[plotMeanAbsTime12Approx, plotMeanAbsTime12Ex]
On[NIntegrate::zeroregion];
On[NIntegrate::inumr];
On[NIntegrate::slwcon];
On[NIntegrate::ncvb];
On[Power::infy];
On[NIntegrate::inumri];
On[Infinity::indet];
```



```
plotMeanAbsTimeD =
 Labeled[Show[plotMeanAbsTime10, plotMeanAbsTime11, plotMeanAbsTime12],
  "D", {{Top, Left}}, FrameMargins → {{14, 0}, {0, 0}},
  LabelStyle → {Directive[FontSize → 18, Bold], FontFamily → "Helvetica"}]
```

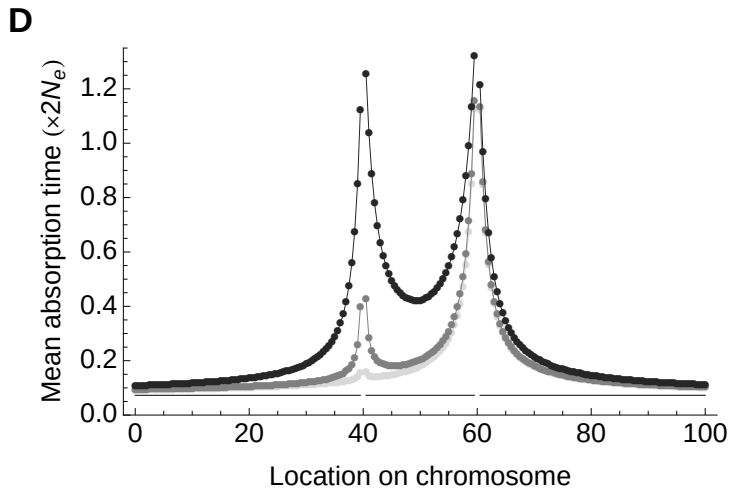
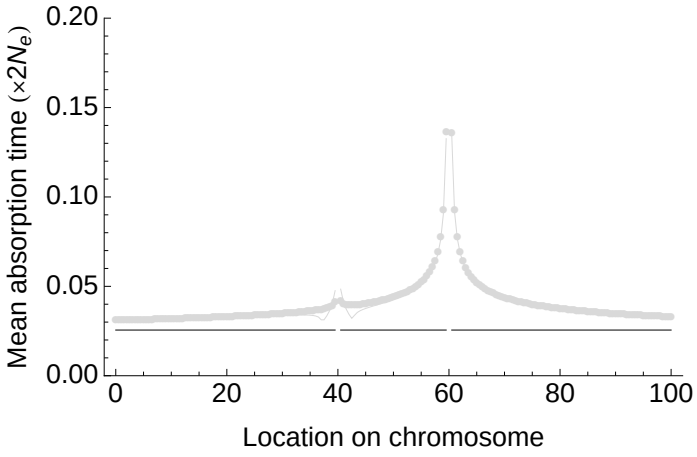


Figure 4: The effect of linkage to two selected sites on the mean absorption time of a neutral de-novo mutation. The loci $\mathcal{A}$ and $\mathcal{B}$ under selection are located 40 and 60 map units from the left end of the chromosome, respectively. It is arbitrarily assumed that one map unit (cM) corresponds to a recombination rate of and the effective size of the island population is $N_e = 100$. The scaled selection coefficient in favour of B_1 is $\beta = 2\,N_e\,b = 80$ and the scaled migration rate is $\mu = 2\,N_e\,m = 0.2$. From light to dark, α/β takes values of 0.005, 0.05 and 0.5, where $\alpha = 2\,N_e\,a$ is the scaled selection coefficient in favour of A_1. Points show values computed using the approximate effective migration rates in equations (1) to (3) and lines are based on numerically computed exact effective migration rates. The horizontal black line denotes the baseline for no linkage to selected sites.

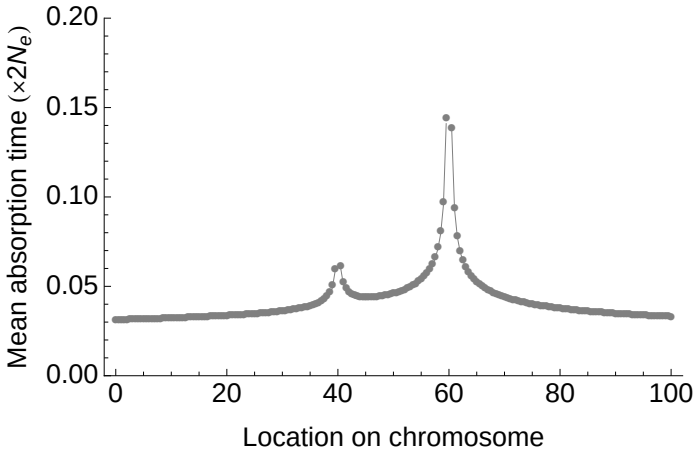
```
Export[PlotPath <> "meanAbsTime_neutrSite_D.eps", plotMeanAbsTimeD, "EPS"]
```

```
/Users/Simon/Documents/LocAdD/results/130419/linkedNeutrSite/figures/
  meanAbsTime_neutrSite_D.eps
```

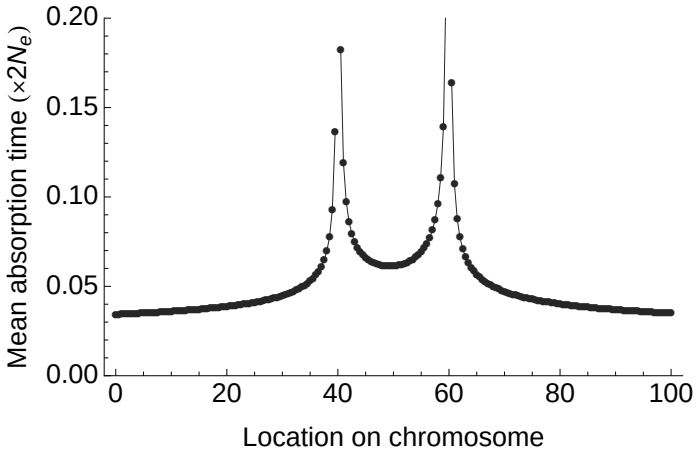
```
mya = 0.002;
myb = .4;
mym = 0.024;
myNe = 100;
myxA = 40;
myxB = 60;
myOffL = 40;
myOffR = 40;
sf = 0.01; (* r = 1 corresponds to 1/sf map units;
one map unit is equivalent to a recombination rate of sf. *)
dist = 0.5;
myRatio = 0.025;(*5;*)
yMax = 0.2;
plotCol = GrayLevel[0.85];
xVals = Range[myxA - myOffL, myxB + myOffR, dist];
Off[NIntegrate::zeroregion];
Off[NIntegrate::inumr];
Off[NIntegrate::slwcon];
Off[NIntegrate::ncvb];
Off[Power::infy];
Off[NIntegrate::inumri];
Off[Infinity::indet];
tBaseVals = meanAbsTimeFunc[0, 0, mym, myNe, myxA, myxB, #, sf, 0.0001] & /@ xVals;
tVals = meanAbsTimeFunc[mya, myb, mym, myNe, myxA, myxB, #, sf, myRatio] & /@ xVals;
tValsEx = meanAbsTimeExactFunc[mya, myb, mym, myNe, myxA, myxB, #, sf][[1]] & /@ xVals;
plotMeanAbsTime13Approx = ListPlot[{tVals},
   DataRange → {0, (myxB + myOffR + (myxA - myOffL))}, PlotRange → {Automatic, {0, yMax}},
   PlotMarkers → {{●, Tiny}, {""}}, PlotStyle → {plotCol}, Joined → False(*,Epilog→
    {{Dashed,Line[{{myxA,0},{myxA,10}}]},{Dashed,Line[{{myxB,0},{myxB,10}}]}}*),
   Frame → True, FrameStyle → {{Black, Opacity[0]}, {Black, Opacity[0]}},
   LabelStyle → {Directive[FontSize → 14], FontFamily → "Helvetica"},
   FrameLabel → {"Location on chromosome", "Mean absorption time (×2N_e)"}];
plotMeanAbsTime13Ex = ListPlot[{tValsEx, tBaseVals},
   DataRange → {0, (myxB + myOffR + (myxA - myOffL))}, PlotRange → {Automatic, {0, yMax}}
   (*,PlotMarkers→{{●,Tiny},{""}}*), PlotStyle → {plotCol, Black}, Joined → True,
   Frame → True, FrameStyle → {{Black, Opacity[0]}, {Black, Opacity[0]}}];
plotMeanAbsTime13 = Show[plotMeanAbsTime13Approx, plotMeanAbsTime13Ex]
On[NIntegrate::zeroregion];
On[NIntegrate::inumr];
On[NIntegrate::slwcon];
On[NIntegrate::ncvb];
On[Power::infy];
On[NIntegrate::inumri];
On[Infinity::indet];
```



```
mya = 0.02;
myb = .4;
mym = 0.024;
myNe = 100;
myxA = 40;
myxB = 60;
myOffL = 40;
myOffR = 40;
sf = 0.01; (* r = 1 corresponds to 1/sf map units;
one map unit is equivalent to a recombination rate of sf. *)
dist = 0.5;
myRatio = 0.025;(*5;*)
yMax = 0.2;
plotCol = GrayLevel[0.5];
xVals = Range[myxA - myOffL, myxB + myOffR, dist];
Off[NIntegrate::zeroregion];
Off[NIntegrate::inumr];
Off[NIntegrate::slwcon];
Off[NIntegrate::ncvb];
Off[Power::infy];
Off[NIntegrate::inumri];
Off[Infinity::indet];
tBaseVals = meanAbsTimeFunc[0, 0, mym, myNe, myxA, myxB, #, sf, 0.0001] & /@ xVals;
tVals = meanAbsTimeFunc[mya, myb, mym, myNe, myxA, myxB, #, sf, myRatio] & /@ xVals;
tValsEx = meanAbsTimeExactFunc[mya, myb, mym, myNe, myxA, myxB, #, sf][[1]] & /@ xVals;
plotMeanAbsTime14Approx = ListPlot[{tVals},
    DataRange → {0, (myxB + myOffR + (myxA - myOffL))}, PlotRange → {Automatic, {0, yMax}},
    PlotMarkers → {{●, Tiny}, {""}}, PlotStyle → {plotCol}, Joined → False(*,Epilog→
     {{Dashed,Line[{{myxA,0},{myxA,10}}]},{Dashed,Line[{{myxB,0},{myxB,10}}]}}*),
    Frame → True, FrameStyle → {{Black, Opacity[0]}, {Black, Opacity[0]}},
    LabelStyle → {Directive[FontSize → 14], FontFamily → "Helvetica"},
    FrameLabel → {"Location on chromosome", "Mean absorption time (×2N_e)"}];
plotMeanAbsTime14Ex = ListPlot[{tValsEx},
    DataRange → {0, (myxB + myOffR + (myxA - myOffL))}, PlotRange → {Automatic, {0, yMax}}
    (*,PlotMarkers→{{●,Tiny},{""}}*), PlotStyle → {plotCol}, Joined → True,
    Frame → True, FrameStyle → {{Black, Opacity[0]}, {Black, Opacity[0]}}];
plotMeanAbsTime14 = Show[plotMeanAbsTime14Approx, plotMeanAbsTime14Ex]
On[NIntegrate::zeroregion];
On[NIntegrate::inumr];
On[NIntegrate::slwcon];
On[NIntegrate::ncvb];
On[Power::infy];
On[NIntegrate::inumri];
On[Infinity::indet];
```



```
mya = 0.2;
myb = .4;
mym = 0.024;
myNe = 100;
myxA = 40;
myxB = 60;
myOffL = 40;
myOffR = 40;
sf = 0.01; (* r = 1 corresponds to 1/sf map units;
one map unit is equivalent to a recombination rate of sf. *)
dist = 0.5;
myRatio = 0.025;(*5;*)
yMax = 0.2;
plotCol = GrayLevel[0.15];
xVals = Range[myxA - myOffL, myxB + myOffR, dist];
Off[NIntegrate::zeroregion];
Off[NIntegrate::inumr];
Off[NIntegrate::slwcon];
Off[NIntegrate::ncvb];
Off[Power::infy];
Off[NIntegrate::inumri];
Off[Infinity::indet];
tBaseVals = meanAbsTimeFunc[0, 0, mym, myNe, myxA, myxB, #, sf, 0.0001] & /@ xVals;
tVals = meanAbsTimeFunc[mya, myb, mym, myNe, myxA, myxB, #, sf, myRatio] & /@ xVals;
tValsEx = meanAbsTimeExactFunc[mya, myb, mym, myNe, myxA, myxB, #, sf][[1]] & /@ xVals;
plotMeanAbsTime15Approx = ListPlot[{tVals},
   DataRange → {0, (myxB + myOffR + (myxA - myOffL))}, PlotRange → {Automatic, {0, yMax}},
   PlotMarkers → {{●, Tiny}, {""}}, PlotStyle → {plotCol}, Joined → False(*,Epilog→
    {{Dashed,Line[{{myxA,0},{myxA,10}}]},{Dashed,Line[{{myxB,0},{myxB,10}}]}}*),
   Frame → True, FrameStyle → {{Black, Opacity[0]}, {Black, Opacity[0]}},
   LabelStyle → {Directive[FontSize → 14], FontFamily → "Helvetica"},
   FrameLabel → {"Location on chromosome", "Mean absorption time (×2N_e)"}];
plotMeanAbsTime15Ex = ListPlot[{tValsEx},
   DataRange → {0, (myxB + myOffR + (myxA - myOffL))}, PlotRange → {Automatic, {0, yMax}}
   (*,PlotMarkers→{{●,Tiny},{""}}*), PlotStyle → {plotCol}, Joined → True,
   Frame → True, FrameStyle → {{Black, Opacity[0]}, {Black, Opacity[0]}}];
plotMeanAbsTime15 = Show[plotMeanAbsTime15Approx, plotMeanAbsTime15Ex]
On[NIntegrate::zeroregion];
On[NIntegrate::inumr];
On[NIntegrate::slwcon];
On[NIntegrate::ncvb];
On[Power::infy];
On[NIntegrate::inumri];
On[Infinity::indet];
```



```
plotMeanAbsTimeE =
 Labeled[Show[plotMeanAbsTime13, plotMeanAbsTime14, plotMeanAbsTime15],
  "E", {{Top, Left}}, FrameMargins → {{14, 0}, {0, 0}},
  LabelStyle → {Directive[FontSize → 18, Bold], FontFamily → "Helvetica"}]
```

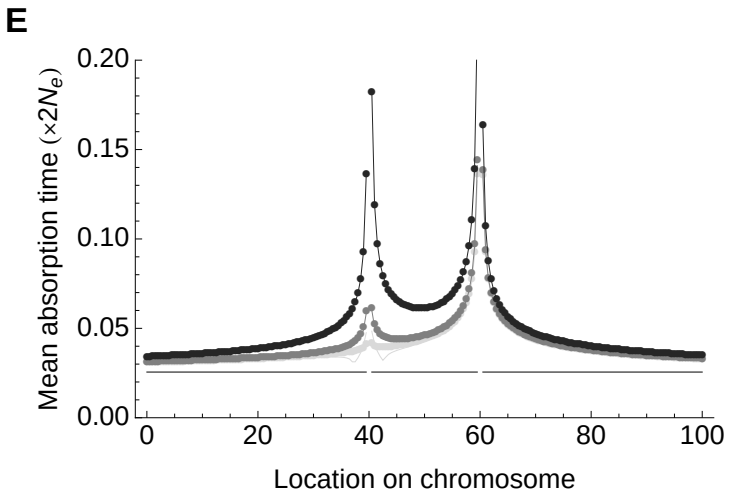
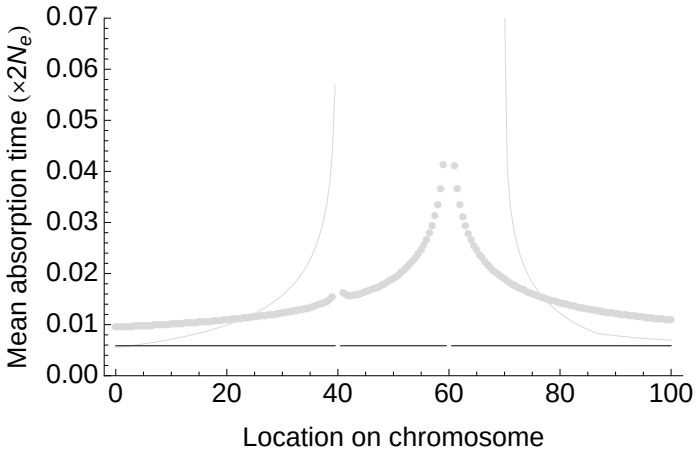


Figure 2: The effect of linkage to two selected sites on the mean absorption time of a neutral de-novo mutation. The loci $\mathcal{A}$ and $\mathcal{B}$ under selection are located 40 and 60 map units from the left end of the chromosome, respectively. It is arbitrarily assumed that one map unit (cM) corresponds to a recombination rate of $r = 0.01$ and the effective size of the island population is $N_e = 100$. The scaled selection coefficient in favour of B_1 is $\beta = 2 N_e b = 80$ and the scaled migration rate is $\mu = 2 N_e m = 4.8$. From light to dark, α / β takes values of 0.005, 0.05 and 0.5, where $\alpha = 2 N_e a$ is the scaled selection coefficient in favour of A_1. Points show values computed using the approximate effective migration rates in equations (1) to (3) and lines are based on numerically computed exact effective migration rates. The horizontal black line denotes the baseline for no linkage to selected sites.

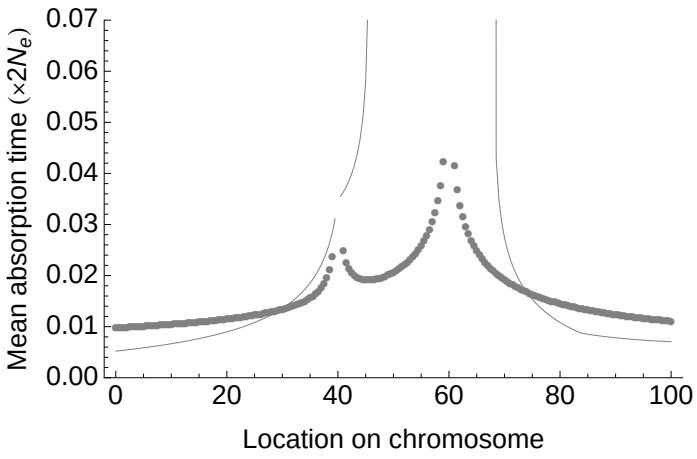
```
Export[PlotPath <> "meanAbsTime_neutrSite_E.eps", plotMeanAbsTimeE, "EPS"]
```

```
/Users/Simon/Documents/LocAdD/results/130419/linkedNeutrSite/figures/
  meanAbsTime_neutrSite_E.eps
```

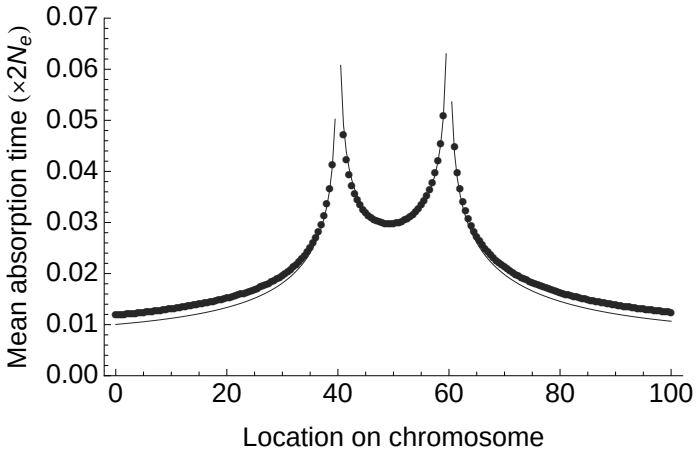
```
mya = 0.002;
myb = .4;
mym = 0.24;
myNe = 100;
myxA = 40;
myxB = 60;
myOffL = 40;
myOffR = 40;
sf = 0.01; (* r = 1 corresponds to 1/sf map units;
one map unit is equivalent to a recombination rate of sf. *)
dist = 0.5;
myRatio = 0.025;(*5;*)
yMax = 0.07;
plotCol = GrayLevel[0.85];
xVals = Range[myxA - myOffL, myxB + myOffR, dist];
Off[NIntegrate::zeroregion];
Off[NIntegrate::inumr];
Off[NIntegrate::slwcon];
Off[NIntegrate::ncvb];
Off[Power::infy];
Off[NIntegrate::inumri];
Off[Infinity::indet];
tBaseVals = meanAbsTimeFunc[0, 0, mym, myNe, myxA, myxB, #, sf, 0.0001] & /@ xVals;
tVals = meanAbsTimeFunc[mya, myb, mym, myNe, myxA, myxB, #, sf, myRatio] & /@ xVals;
tValsEx = meanAbsTimeExactFunc[mya, myb, mym, myNe, myxA, myxB, #, sf][[1]] & /@ xVals;
plotMeanAbsTime16Approx = ListPlot[{tVals},
   DataRange → {0, (myxB + myOffR + (myxA - myOffL))}, PlotRange → {Automatic, {0, yMax}},
   PlotMarkers → {{●, Tiny}, {""}}, PlotStyle → {plotCol}, Joined → False(*,Epilog→
    {{Dashed,Line[{{myxA,0},{myxA,10}}]},{Dashed,Line[{{myxB,0},{myxB,10}}]}}*),
   Frame → True, FrameStyle → {{Black, Opacity[0]}, {Black, Opacity[0]}},
   LabelStyle → {Directive[FontSize → 14], FontFamily → "Helvetica"},
   FrameLabel → {"Location on chromosome", "Mean absorption time (×2N_e)"}];
plotMeanAbsTime16Ex = ListPlot[{tValsEx, tBaseVals},
   DataRange → {0, (myxB + myOffR + (myxA - myOffL))}, PlotRange → {Automatic, {0, yMax}}
   (*,PlotMarkers→{{●,Tiny},{""}}*), PlotStyle → {plotCol, Black}, Joined → True,
   Frame → True, FrameStyle → {{Black, Opacity[0]}, {Black, Opacity[0]}}];
plotMeanAbsTime16 = Show[plotMeanAbsTime16Approx, plotMeanAbsTime16Ex]
On[NIntegrate::zeroregion];
On[NIntegrate::inumr];
On[NIntegrate::slwcon];
On[NIntegrate::ncvb];
On[Power::infy];
On[NIntegrate::inumri];
On[Infinity::indet];
```



```
mya = 0.02;
myb = .4;
mym = 0.24;
myNe = 100;
myxA = 40;
myxB = 60;
myOffL = 40;
myOffR = 40;
sf = 0.01; (* r = 1 corresponds to 1/sf map units;
one map unit is equivalent to a recombination rate of sf. *)
dist = 0.5;
myRatio = 0.025;(*5;*)
yMax = 0.07;
plotCol = GrayLevel[0.5];
xVals = Range[myxA - myOffL, myxB + myOffR, dist];
Off[NIntegrate::zeroregion];
Off[NIntegrate::inumr];
Off[NIntegrate::slwcon];
Off[NIntegrate::ncvb];
Off[Power::infy];
Off[NIntegrate::inumri];
Off[Infinity::indet];
tBaseVals = meanAbsTimeFunc[0, 0, mym, myNe, myxA, myxB, #, sf, 0.0001] & /@ xVals;
tVals = meanAbsTimeFunc[mya, myb, mym, myNe, myxA, myxB, #, sf, myRatio] & /@ xVals;
tValsEx = meanAbsTimeExactFunc[mya, myb, mym, myNe, myxA, myxB, #, sf][[1]] & /@ xVals;
plotMeanAbsTime17Approx = ListPlot[{tVals},
    DataRange → {0, (myxB + myOffR + (myxA - myOffL))}, PlotRange → {Automatic, {0, yMax}},
    PlotMarkers → {{●, Tiny}, {""}}, PlotStyle → {plotCol}, Joined → False(*,Epilog→
     {{Dashed,Line[{{myxA,0},{myxA,10}}]},{Dashed,Line[{{myxB,0},{myxB,10}}]}}*),
    Frame → True, FrameStyle → {{Black, Opacity[0]}, {Black, Opacity[0]}},
    LabelStyle → {Directive[FontSize → 14], FontFamily → "Helvetica"},
    FrameLabel → {"Location on chromosome", "Mean absorption time (×2N_e)"}];
plotMeanAbsTime17Ex = ListPlot[{tValsEx},
    DataRange → {0, (myxB + myOffR + (myxA - myOffL))}, PlotRange → {Automatic, {0, yMax}}
    (*,PlotMarkers→{{●,Tiny},{""}}*), PlotStyle → {plotCol}, Joined → True,
    Frame → True, FrameStyle → {{Black, Opacity[0]}, {Black, Opacity[0]}}];
plotMeanAbsTime17 = Show[plotMeanAbsTime17Approx, plotMeanAbsTime17Ex]
On[NIntegrate::zeroregion];
On[NIntegrate::inumr];
On[NIntegrate::slwcon];
On[NIntegrate::ncvb];
On[Power::infy];
On[NIntegrate::inumri];
On[Infinity::indet];
```



```
mya = 0.2;
myb = .4;
mym = 0.24;
myNe = 100;
myxA = 40;
myxB = 60;
myOffL = 40;
myOffR = 40;
sf = 0.01; (* r = 1 corresponds to 1/sf map units;
one map unit is equivalent to a recombination rate of sf. *)
dist = 0.5;
myRatio = 0.025;(*5;*)
yMax = 0.07;
plotCol = GrayLevel[0.15];
xVals = Range[myxA - myOffL, myxB + myOffR, dist];
Off[NIntegrate::zeroregion];
Off[NIntegrate::inumr];
Off[NIntegrate::slwcon];
Off[NIntegrate::ncvb];
Off[Power::infy];
Off[NIntegrate::inumri];
Off[Infinity::indet];
tBaseVals = meanAbsTimeFunc[0, 0, mym, myNe, myxA, myxB, #, sf, 0.0001] & /@ xVals;
tVals = meanAbsTimeFunc[mya, myb, mym, myNe, myxA, myxB, #, sf, myRatio] & /@ xVals;
tValsEx = meanAbsTimeExactFunc[mya, myb, mym, myNe, myxA, myxB, #, sf][[1]] & /@ xVals;
plotMeanAbsTime18Approx = ListPlot[{tVals},
    DataRange → {0, (myxB + myOffR + (myxA - myOffL))}, PlotRange → {Automatic, {0, yMax}},
    PlotMarkers → {{●, Tiny}, {""}}, PlotStyle → {plotCol}, Joined → False(*,Epilog→
     {{Dashed,Line[{{myxA,0},{myxA,10}}]},{Dashed,Line[{{myxB,0},{myxB,10}}]}}*),
    Frame → True, FrameStyle → {{Black, Opacity[0]}, {Black, Opacity[0]}},
    LabelStyle → {Directive[FontSize → 14], FontFamily → "Helvetica"},
    FrameLabel → {"Location on chromosome", "Mean absorption time (×2N_e)"}];
plotMeanAbsTime18Ex = ListPlot[{tValsEx},
    DataRange → {0, (myxB + myOffR + (myxA - myOffL))}, PlotRange → {Automatic, {0, yMax}}
    (*,PlotMarkers→{{●,Tiny},{""}}*), PlotStyle → {plotCol}, Joined → True,
    Frame → True, FrameStyle → {{Black, Opacity[0]}, {Black, Opacity[0]}}];
plotMeanAbsTime18 = Show[plotMeanAbsTime18Approx, plotMeanAbsTime18Ex]
On[NIntegrate::zeroregion];
On[NIntegrate::inumr];
On[NIntegrate::slwcon];
On[NIntegrate::ncvb];
On[Power::infy];
On[NIntegrate::inumri];
On[Infinity::indet];
```



Remark: It is arbitrarily assumed here that one map unit corresponds to a recombination rate of $r = 0.01$.

```
plotMeanAbsTimeF =
 Labeled[Show[plotMeanAbsTime16, plotMeanAbsTime17, plotMeanAbsTime18],
  "F", {{Top, Left}}, FrameMargins → {{14, 0}, {0, 0}},
  LabelStyle → {Directive[FontSize → 18, Bold], FontFamily → "Helvetica"}]
```

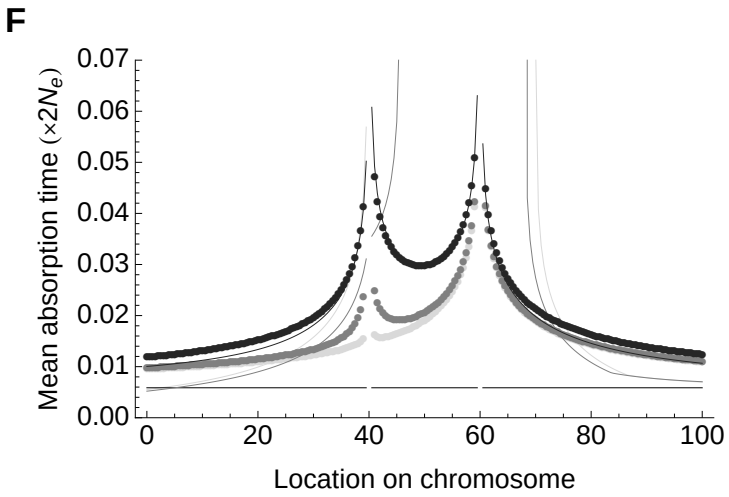


Figure 6: The effect of linkage to two selected sites on the mean absorption time of a neutral de-novo mutation. The biallelic loci $\mathcal{A}$ and $\mathcal{B}$ under selection are located 40 and 60 map units from the left end of the chromosome, respectively. It is arbitrarily assumed that one map unit corresponds to a recombination rate of $r = 0.01$ and the effective size of the island population is $N_e = 100$. The scaled selection coefficient in favour of B_1 is $\beta = 2\,N_e\,b = 80$ and the scaled migration rate is $\mu = 2\,N_e\,m = 48$. From light to dark, α/β takes values of 0.005, 0.05 and 0.5, where $\alpha = 2\,N_e\,a$ is the scaled selection coefficient in favour of A_1. Points show values computed using the approximate effective migration rates in equations (1) to (3) and lines are based on numerically computed exact effective migration rates. The horizontal black line denotes the baseline for no linkage to selected sites.

Remark: It is arbitrarily assumed here that one map unit corresponds to a recombination rate of $r = 0.01$.

```
Export[PlotPath <> "meanAbsTime_neutrSite_F.eps", plotMeanAbsTimeF, "EPS"]
```

```
/Users/Simon/Documents/LocAdD/results/130419/linkedNeutrSite/figures/
  meanAbsTime_neutrSite_F.eps
```

■ Stationary distribution of allele frequencies for $n_c > 0$

■ Derivation starting from the forward Kolmogorov equation

To obtain the stationary distribution of n, we follow Ewens (1979; pp. 145–146). Starting from the backward Kolmogorov equation, integration with respect to n shows that the rate of flow of probability across the point n at time t is given by

$$\frac{\partial}{\partial t}[1 - F(n;t)] = M(n)\,f(n;t) - \frac{1}{2}\,\frac{\partial\{V(n)\,f(n;t)\}}{\partial x}, \tag{25}$$

where $F(n;t)$ is the (cumulative) distribution function

$$F(n;t) = \int_0^n f(y;t)\,dy. \tag{26}$$

As above, $M(n)$ and $V(n)$ are the mean and variance of the change in n over a small amount of time, δt.

If a stationary distribution $f(n)$ exists, the probability flux will be zero if $f(n;t)$ in Eq. (25) is replaced by $f(n)$, so that the stationary distribution satisfies

$$-M(n)\,f(n) + \frac{1}{2}\,\frac{d\{V(n)\,f(n)\}}{dn} = 0 \Leftrightarrow \frac{d\{V(n)\,f(n)\}}{dn} = 2\,M(n)\,f(n). \tag{27}$$

Dividing both sides by $V(n)\,f(n)$ and multiplying by dn, we obtain

$$2\,\frac{M(n)}{V(n)}\,dn = \frac{1}{V(n)\,f(n)}\,d\{V(n)\,f(n)\}. \tag{28}$$

Substituting $g(n)$ for $V(n)\,f(n)$, this is

$$2\,\frac{M(n)}{V(n)}\,dn = \frac{1}{g(n)}\,dg(n). \tag{29}$$

Integrating both sides and adjusting variable names yields

$$\log[g(n)] = \log[V(n)\, f(n)] = 2 \int^{n} \frac{M(y)}{V(y)}\, dy + C_1, \tag{30}$$

where C_1 is a constant of integration. Taking to the power of e and rearranging, we obtain the solution, up to a constant C_2, as

$$f(n) = \frac{C_2}{V(n)}\, e^{2 \int^{n} \frac{M(y)}{V(y)}\, dy}. \tag{31}$$

The constant must be defined such that

$$\int_0^1 f(n)\, dn = 1. \tag{32}$$

Recalling Eq. (13) and (14) for $M(n)$ and $V(n)$, we find

$$f(n) = \frac{C_2}{n(1-n)}\, e^{2 \int^{n} \frac{\mu(n_c - y)}{y(1-y)}\, dy} = C_2\, n^{2 n_c \mu - 1} (1-n)^{2(1-n_c)\mu - 1}. \tag{33}$$

We are left with determining C_2, which is achieved by integrating Eq. (33) over n from 0 to 1 and assigning the inverse of this integral to C_2. This yields the well-known result

$$f(n) = \frac{\Gamma(2\,\mu)}{\Gamma(2\,\mu\, n_c)\, \Gamma(2\,\mu(1-n_c))}\, n^{2\,\mu\, n_c - 1} (1-n)^{2\,\mu(1-n_c) - 1} \tag{34}$$

with diffusion-scaled parameters, or

$$f(n) = \frac{\Gamma(4\, N_e\, m)}{\Gamma(4\, N_e\, n_c\, m)\, \Gamma(4\, N_e\, m(1-n_c))}\, n^{4\, N_e\, m\, n_c - 1} (1-n)^{4\, N_e\, m(1-n_c) - 1} \tag{35}$$

for unscaled parameters (*cf.* Wright 1940, p. 239–241), where $\Gamma(x)$ is the Gamma function.

```
2 Integrate[μ (nC - n)/(n (1 - n)), n] // FullSimplify
```

```
2 μ (-(-1 + nC) Log[1 - n] + nC Log[n])
```

```
FullSimplify[e^(2 Integrate[μ (nC-n)/(n (1-n)), n])]
```

```
(1 - n)^(-2 (-1+nC) μ) n^(2 nC μ)
```

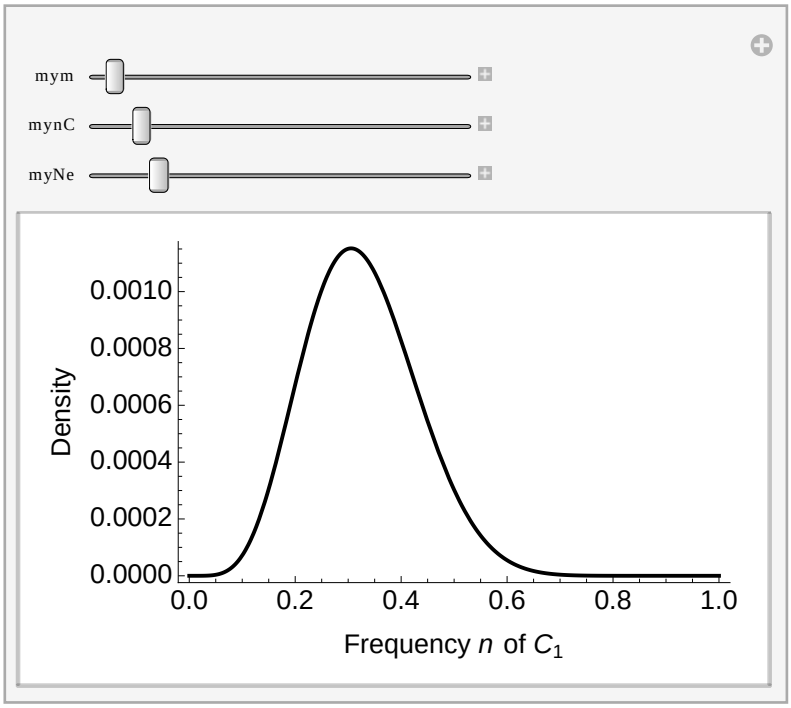
```
Integrate[n^(2 nC μ-1) (1 - n)^(2 (1-nC) μ-1), {n, 0, 1}, Assumptions → {0 < nC < 1, 0 < μ}]
```

$$\frac{\text{Gamma}[-2\,(-1+\text{nC})\,\mu]\ \text{Gamma}[2\,\text{nC}\,\mu]}{\text{Gamma}[2\,\mu]}$$

In[236]:=
```
ruleC2 := C2 → 1 / (Gamma[-2 (-1 + nC) μ] Gamma[2 nC μ] / Gamma[2 μ])
```

In[237]:=
```
statFreqDist[m_, nC_, Ne_, n_] :=
  Gamma[4 Ne m] / (Gamma[4 Ne m nC] Gamma[4 Ne m (1 - nC)]) n^(4 Ne m nC-1) (1 - n)^(4 Ne m (1-nC)-1)
```

```
Manipulate[Plot[statFreqDist[mym, mynC, myNe, n], {n, 0, 1},
  Frame → True, FrameStyle → {{Black, Opacity[0]}, {Black, Opacity[0]}},
  LabelStyle → {Directive[FontSize → 14], FontFamily → "Helvetica"},
  FrameLabel → {"Frequency n of C₁", "Density"}, PlotStyle → {Thick, Black},
  PlotRange → Full], {{mym, 0.024}, 0, 1}, {{mynC, 0.1}, 0, 1}, {{myNe, 150}, 0, 1000}]
```



Beta distribution

- Mean

```
FullSimplify[Integrate[x Gamma[α + β]/(Gamma[α] Gamma[β]) x^(α-1) (1 - x)^(β-1), {x, 0, 1}],
  Assumptions → {Re[β] > 0, Re[α] > -1}]
```

$$\frac{\alpha}{\alpha+\beta}$$

- Mean square

```
FullSimplify[Integrate[x^2 Gamma[α + β]/(Gamma[α] Gamma[β]) x^(α-1) (1 - x)^(β-1), {x, 0, 1}],
  Assumptions → {Re[β] > 0, Re[α] > -1}]
```

$$\frac{\alpha\,(1+\alpha)}{(\alpha+\beta)\,(1+\alpha+\beta)}$$

- Variance

```
varBeta = α (1 + α)/((α + β) (1 + α + β)) - (α/(α + β))^2 // FullSimplify
```

$$\frac{\alpha\,\beta}{(\alpha+\beta)^2\,(1+\alpha+\beta)}$$

```
% /. {α → 4 Ne me nC, β → 4 Ne me (1 - nC)} // FullSimplify
```

$$-\frac{(-1+nC)\,nC}{1+4\,me\,Ne}$$

This is in agreement with Petry (1983, Eq. 4).

- Heterozygosity

```
FullSimplify[Integrate[2 x (1 - x) Gamma[α + β]/(Gamma[α] Gamma[β]) x^(α-1) (1 - x)^(β-1), {x, 0, 1}],
 Assumptions → {Re[β] > 0, Re[α] > -1}]
```

$$\frac{2\,\alpha\,\beta}{(\alpha+\beta)\,(1+\alpha+\beta)}$$

Check via homozygosity:

```
FullSimplify[Integrate[(1 - x)^2 Gamma[α + β]/(Gamma[α] Gamma[β]) x^(α-1) (1 - x)^(β-1), {x, 0, 1}],
 Assumptions → {Re[β] > 0, Re[α] > 0}]
```

$$\frac{\beta\,(1+\beta)}{(\alpha+\beta)\,(1+\alpha+\beta)}$$

```
FullSimplify[Integrate[(x)^2 Gamma[α + β]/(Gamma[α] Gamma[β]) x^(α-1) (1 - x)^(β-1), {x, 0, 1}],
 Assumptions → {Re[β] > 0, Re[α] > 0}]
```

$$\frac{\alpha\,(1+\alpha)}{(\alpha+\beta)\,(1+\alpha+\beta)}$$

```
Hom = β (1 + β)/((α + β) (1 + α + β)) + α (1 + α)/((α + β) (1 + α + β)) // FullSimplify
```

$$\frac{\alpha+\alpha^2+\beta+\beta^2}{(\alpha+\beta)\,(1+\alpha+\beta)}$$

```
Het = 1 - Hom // FullSimplify
```

$$\frac{2\,\alpha\,\beta}{(\alpha+\beta)\,(1+\alpha+\beta)}$$

```
Het /. {{α → 4 Ne me nC, β → 4 Ne me (1 - nC)}} // FullSimplify
```

$$\left\{-\frac{8\,me\,(-1+nC)\,nC\,Ne}{1+4\,me\,Ne}\right\}$$

- F_{ST}

```
FullSimplify[
 Integrate[(2 nC (1 - nC) - 2 x (1 - x))/(2 nC (1 - nC)) Gamma[α + β]/(Gamma[α] Gamma[β]) x^(α-1) (1 - x)^(β-1), {x, 0, 1}],
 Assumptions → {Re[β] > 0, Re[α] > 0}]
```

$$\frac{-1+nC+\frac{\alpha\,\beta}{nC\,(\alpha+\beta)\,(1+\alpha+\beta)}}{-1+nC}$$

```
% /. {α → 4 Ne me nC, β → 4 Ne me (1 - nC)} // FullSimplify
```

$$\frac{1}{1+4\,me\,Ne}$$

Alternatively, $F_{ST} = \frac{\mathrm{var}(n_c)}{n_c(1-n_c)}$:

$$\frac{\alpha\,\beta}{(\alpha+\beta)^2\,(1+\alpha+\beta)} \Big/ (nC\,(1-nC))$$ // FullSimplify

$$-\frac{\alpha\,\beta}{(-1+nC)\,nC\,(\alpha+\beta)^2\,(1+\alpha+\beta)}$$

% /. {α → 4 Ne me nC, β → 4 Ne me (1 − nC)} // FullSimplify

$$\frac{1}{1+4\,me\,Ne}$$

■ Application: insertion of effective migration rates

Substituting the effective migration rates for the constellations $\mathcal{A}$-C-$\mathcal{B}$, $\mathcal{A}$-$\mathcal{B}$-C and C-$\mathcal{A}$-$\mathcal{B}$ from equations (2) – (3) for m in the stationary distribution (34) yields the effective stationary distribution for a linked neutral locus C as a function of the distance to the two loci under selection ($\mathcal{A}$ and $\mathcal{B}$). As the resulting expressions cannot be further simplified analytically, we directly proceed with a graphical exploration.

statFreqDist[m, nC, Ne, n] /. ruleMeABC // FullSimplify

$$\left(\left(-(-1+n)\,n^{-1+\frac{4\,m\,nC\,Ne\,rBC\,(b+rAB+rBC)}{(b+rBC)\,(a+b+rAB+rBC)}}\right)^{-1-\frac{4\,m\,(-1+nC)\,Ne\,rBC\,(b+rAB+rBC)}{(b+rBC)\,(a+b+rAB+rBC)}}\,\text{Gamma}\left[\frac{4\,m\,Ne\,rBC\,(b+rAB+rBC)}{(b+rBC)\,(a+b+rAB+rBC)}\right]\right) \Big/$$
$$\left(\text{Gamma}\left[-\frac{4\,m\,(-1+nC)\,Ne\,rBC\,(b+rAB+rBC)}{(b+rBC)\,(a+b+rAB+rBC)}\right]\text{Gamma}\left[\frac{4\,m\,nC\,Ne\,rBC\,(b+rAB+rBC)}{(b+rBC)\,(a+b+rAB+rBC)}\right]\right)$$

In[238]:=

```
statDistFunc::usage =
  "statDistFunc[a, b, m, Ne, xA, xB, xC, scaleFac] returns the value of the
    stationary distribution (density) for a neutral de-novo mutation that is
    linked to two sites under selection, valid for weak migration compared
    to selection. The selection coefficients at the two sites are a and b,
    the actual migration rate is m and the positions of the three sites in
    map units are xA, xB and xC, where the latter belongs to the neutral
    locus and xA < xB is assumed without loss of generality. The scaleFac
    defines what recombination rate should correspond to one map unit.";
statDistFunc[a_, b_, m_, nC_, Ne_, n_, xA_, xB_, xC_, scaleFac_] :=
 Module[{me, statDist, rAC, rBC},
  rAC = Abs[xA - xC] scaleFac;
  rBC = Abs[xB - xC] scaleFac;
  me = If[xC < xA < xB, m rAC (a + rBC)/((a + rAC) (a + b + rBC)), If[xA ≤ xC ≤ xB,
      m rAC rBC/((a + rAC) (b + rBC)), If[xA < xB < xC, m rBC (b + rAC)/((b + rBC) (a + b + rAC)), Null]]];
  statDist = Gamma[4 Ne me]/(Gamma[4 Ne me nC] Gamma[4 Ne me (1 - nC)]
    n^(4 Ne me nC - 1) * (1 - n)^(4 Ne me (1 - nC) - 1);
  Return[If[xA == xC, Null, If[xB == xC, Null, statDist]]]
  (* We return 'Null' if xA==xC or xB==xC. *)
 ]
```

In[240]:=

```
statDistFuncWrong::usage =
  "statDistFunc[a, b, m, Ne, xA, xB, xC, scaleFac] returns the value of the
    stationary distribution (density) for a neutral de-novo mutation that is
    linked to two sites under selection, valid for weak migration compared
    to selection. The selection coefficients at the two sites are a and b,
    the actual migration rate is m and the positions of the three sites in
    map units are xA, xB and xC, where the latter belongs to the neutral
    locus and xA < xB is assumed without loss of generality. The scaleFac
    defines what recombination rate should correspond to one map unit.";
statDistFuncWrong[a_, b_, m_, nC_, Ne_, n_, xA_, xB_, xC_, scaleFac_] :=
 Module[{me, statDist, rAC, rBC},
  rAC = Abs[xA - xC] scaleFac;
  rBC = Abs[xB - xC] scaleFac;
  me = If[xC < xA < xB, m rAC (a + rBC) / ((a + rAC) (a + b + rBC)), If[xA ≤ xC ≤ xB,
      m rAC rBC / ((a + rAC) (b + rBC)), If[xA < xB < xC, m rBC (b + rAC) / ((b + rBC) (a + b + rAC)), Null]]];
  statDist = Gamma[4 Ne me] / (Gamma[4 Ne me nC] Gamma[4 Ne me (1 - nC)]) n^(4 Ne me nC - 1) (1 - n)^(4 Ne me (1 - nC) - 1);
  Return[If[xA == xC, Null, If[xB == xC, Null, statDist]]]
  (* We return 'Null' if xA==xC or xB==xC. *)
 ]
```

In[242]:=

```
effMigRateFunc::usage =
  "effMigRateFunc[a, b, m, Ne, xA, xB, xC, scaleFac] returns the effective
    migration rate m_e for a neutral de-novo mutation that is linked to
    two sites under selection, valid for weak migration compared to
    selection. The selection coefficients at the two sites are a and b,
    the actual migration rate is m and the positions of the three sites in
    map units are xA, xB and xC, where the latter belongs to the neutral
    locus and xA < xB is assumed without loss of generality. The scaleFac
    defines what recombination rate should correspond to one map unit.";
effMigRateFunc[a_, b_, m_, Ne_, xA_, xB_, xC_, scaleFac_] := Module[{me, rAC, rBC},
  rAC = Abs[xA - xC] scaleFac;
  rBC = Abs[xB - xC] scaleFac;
  me = If[xC < xA < xB, m rAC (a + rBC) / ((a + rAC) (a + b + rBC)), If[xA ≤ xC ≤ xB,
      m rAC rBC / ((a + rAC) (b + rBC)), If[xA < xB < xC, m rBC (b + rAC) / ((b + rBC) (a + b + rAC)), Null]]];
  Return[If[xA == xC, Null, If[xB == xC, Null, me]]]
  (* We return 'Null' if xA==xC or xB==xC. *)
 ]
```

In[244]:=

```
effMigRateExactFunc::usage="effMigRateExactFunc[a, b, m, Ne, xA, xB, xC, scaleFac] ret
effMigRateExactFunc[a_,b_,m_,Ne_,xA_,xB_,xC_,scaleFac_]:=Module[{me,rAC,rBC,JNACB,JNAB
rAC=Abs[xA-xC]scaleFac;
rBC=Abs[xB-xC]scaleFac;
JNACB={{-m,a,b,0},{1/(8 a (r1+r2)) m (-a^2+b^2+6 a (r1+r2)-4 m (r1+r2)-(r1+r2)^2+(a-b+r1+r2) √
JNABC={{-m,a,b,0},{m (-a^2+b^2+6 a r1-4 m r1-r1^2+(a-b+r1) √(-8 m r1+(a+b+r1)^2))/(8 a r1),a-m-r1-
JNCAB={{-m,b,a,0},{m (a^2-b^2+6 b r2-4 m r2-r2^2+(-a+b+r2) √(-8 m r2+(a+b+r2)^2))/(8 b r2),b-m-r1-r2
me=If[xC<xA<xB,-Max[Re[Eigenvalues[JNCAB]]],If[xA≤xC≤xB,-Max[Re[Eigenvalues[JNACB]]],I
Return[If[xA==xC,Null,If[xB==xC,Null,me]]](* We return 'Null' if xA==xC or xB==xC. *)
]
```

In[246]:=

```
FstFunc::usage =
  "FstFunc[a, b, m, Ne, xA, xB, xC, scaleFac, ratio] returns the mean F_ST
    for a neutral de-novo mutation that is linked to two sites under
    selection, valid for weak migration compared to selection. The
    selection coefficients at the two sites are a and b, the actual
    migration rate is m and the positions of the three sites in map units
    are xA, xB and xC, where the latter belongs to the neutral locus and
    xA < xB is assumed without loss of generality. The scaleFac defines
    what recombination rate should correspond to one map unit, and ratio
    denotes by how much migration must be weaker compared to recombination.";
FstFunc[a_, b_, m_, nC_, Ne_, xA_, xB_, xC_, scaleFac_, ratio_] :=
 Module[{me, meanFST, rAC, rBC, x},
  rAC = Abs[xA - xC] scaleFac;
  rBC = Abs[xB - xC] scaleFac;
  x = Min[rAC, rBC];
  me = If[xC < xA < xB, m rAC (a + rBC)/((a + rAC) (a + b + rBC)), If[xA ≤ xC ≤ xB,
     m rAC rBC/((a + rAC) (b + rBC)), If[xA < xB < xC, m rBC (b + rAC)/((b + rBC) (a + b + rAC)), Null]]];
  meanFST = 1/(1 + 4 Ne me);
  Return[If[xA == xC || xB == xC, Null, If[(m ratio ≤ x), Chop[meanFST], Null]]]
  (*Return[If[xA==xC,Null,If[xB==xC,Null,meanFST]]]*)
  (* We return 'Null' if xA==xC or xB==xC. *)
 ]
```

In[248]:=

```
FstExactFunc::usage="FstExactFunc[a, b, m, Ne, xA, xB, xC, scaleFac] returns the mean
FstExactFunc[a_,b_,m_,nC_,Ne_,xA_,xB_,xC_,scaleFac_]:=Module[{me,meanFST,rAC,rBC,JNABC
rAC=Abs[xA-xC]scaleFac;
rBC=Abs[xB-xC]scaleFac;
JNACB={{-m,a,b,0},{1/(8 a (r1+r2)) m (-a^2+b^2+6 a (r1+r2)-4 m (r1+r2)-(r1+r2)^2+(a-b+r1+r2) √
JNABC={{-m,a,b,0},{m (-a^2+b^2+6 a r1-4 m r1-r1^2+(a-b+r1) √(-8 m r1+(a+b+r1)^2))/(8 a r1),a-m-r1-
JNCAB={{-m,b,a,0},{m (a^2-b^2+6 b r2-4 m r2-r2^2+(-a+b+r2) √(-8 m r2+(a+b+r2)^2))/(8 b r2),b-m-r1-r2
me=If[xC<xA<xB,-Max[Re[Eigenvalues[JNCAB]]],If[xA≤xC≤xB,-Max[Re[Eigenvalues[JNACB]]],I
meanFST=1/(1+4 Ne me);
Return[If[xA==xC||xB==xC,Null,Chop[meanFST]]]
]
```

In[250]:=

```
hetFuncOld::usage =
  "hetFunc[a, b, m, Ne, xA, xB, xC, scaleFac, ratio] returns the mean
    heterozygosity H for a neutral de-novo mutation that is linked to two sites
    under selection, valid for weak migration and/or weak recmbination compared
    to selection. The selection coefficients at the two sites are a and b,
    the actual migration rate is m and the positions of the three sites in map
    units are xA, xB and xC, where the latter belongs to the neutral locus
    and xA < xB is assumed without loss of generality. The scaleFac defines
    what recombination rate should correspond to one map unit, and ratio
    denotes by how much migration must be weaker compared to recombination.";
hetFuncOld[a_, b_, m_, nC_, Ne_, xA_, xB_, xC_, scaleFac_, ratio_] :=
 Module[{me, H, rAC, rBC, x},
  rAC = Abs[xA - xC] scaleFac;
  rBC = Abs[xB - xC] scaleFac;
  x = Min[rAC, rBC];
  me = If[xC < xA < xB, m rAC (a + rBC)/((a + rAC) (a + b + rBC)), If[xA ≤ xC ≤ xB,
      m rAC rBC/((a + rAC) (b + rBC)), If[xA < xB < xC, m rBC (b + rAC)/((b + rBC) (a + b + rAC)), Null]]];
  H = Gamma[4 Ne me]/(Gamma[4 Ne me nC] Gamma[4 Ne me (1 - nC)]) *
    NIntegrate[2 n (1 - n) n^(4 Ne me nC - 1) (1 - n)^(4 Ne me (1 - nC) - 1), {n, 0, 1}];
  Return[If[xA == xC || xB == xC, Null, If[(m ratio ≤ x), Chop[H], Null]]]
  (*Return[If[xA==xC,Null,If[xB==xC,Null,H]]]*)
  (* We return 'Null' if xA==xC or xB==xC. *)
 ]
```

A more efficient implementation based on properties of the beta distribution (see also Wright 1940 or Petry 1983).

In[252]:=

```
hetFunc::usage =
  "hetFunc[a, b, m, Ne, xA, xB, xC, scaleFac, ratio] returns the mean
    heterozygosity H for a neutral de-novo mutation that is linked to two sites
    under selection, valid for weak migration and/or weak recmbination compared
    to selection. The selection coefficients at the two sites are a and b,
    the actual migration rate is m and the positions of the three sites in map
    units are xA, xB and xC, where the latter belongs to the neutral locus
    and xA < xB is assumed without loss of generality. The scaleFac defines
    what recombination rate should correspond to one map unit, and ratio
    denotes by how much migration must be weaker compared to recombination.";
hetFunc[a_, b_, m_, nC_, Ne_, xA_, xB_, xC_, scaleFac_, ratio_] :=
 Module[{me, H, rAC, rBC, x},
  rAC = Abs[xA - xC] scaleFac;
  rBC = Abs[xB - xC] scaleFac;
  x = Min[rAC, rBC];
  me = If[xC < xA < xB, m rAC (a + rBC) / ((a + rAC) (a + b + rBC)), If[xA ≤ xC ≤ xB,
      m rAC rBC / ((a + rAC) (b + rBC)), If[xA < xB < xC, m rBC (b + rAC) / ((b + rBC) (a + b + rAC)), Null]]];
  H = 2 * 4 Ne me nC (1 - nC) / (1 + 4 Ne me);
  Return[If[xA == xC || xB == xC, Null, If[(m ratio ≤ x), Chop[H], Null]]]
  (*Return[If[xA==xC,Null,If[xB==xC,Null,H]]]*)
  (* We return 'Null' if xA==xC or xB==xC. *)
 ]
```

```
mya = 0.02;
myb = 0.4;
mym = 0.01;
mynC = 0.2;
myNe = 100;
myxA = 20;
myxB = 60;
myxC = 9999.9999;
mySF = 0.01;
myRatio = 0.0001;

hetFuncOld[mya, myb, mym, mynC, myNe, myxA, myxB, myxC, mySF, myRatio]
hetFunc[mya, myb, mym, mynC, myNe, myxA, myxB, myxC, mySF, myRatio]
```

```
0.255784

0.255784
```

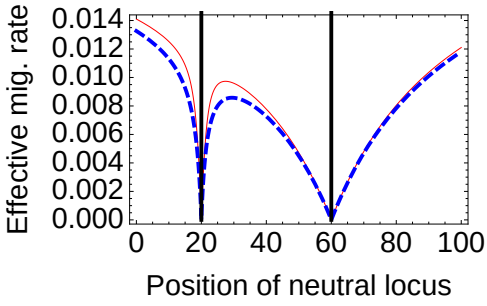
In[254]:=

```
hetExactFunc::usage="hetExactFunc[a, b, m, Ne, xA, xB, xC, scaleFac] returns the mean
hetExactFunc[a_,b_,m_,nC_,Ne_,xA_,xB_,xC_,scaleFac_]:=Module[{me,H,rAC,rBC,JNABC,JNACB
rAC=Abs[xA-xC]scaleFac;
rBC=Abs[xB-xC]scaleFac;
JNACB={{-m,a,b,0},{1/(8 a (r1+r2)) m (-a^2+b^2+6 a (r1+r2)-4 m (r1+r2)-(r1+r2)^2+(a-b+r1+r2) √
JNABC={{-m,a,b,0},{m (-a^2+b^2+6 a r1-4 m r1-r1^2+(a-b+r1) √(-8 m r1+(a+b+r1)^2))/(8 a r1),a-m-r1-
JNCAB={{-m,b,a,0},{m (a^2-b^2+6 b r2-4 m r2-r2^2+(-a+b+r2) √(-8 m r2+(a+b+r2)^2))/(8 b r2),b-m-r1-r2
me=If[xC<xA<xB,-Max[Re[Eigenvalues[JNCAB]]],If[xA≤xC≤xB,-Max[Re[Eigenvalues[JNACB]]],I
(*H=Gamma[4Ne me]/(Gamma[4Ne me nC ]Gamma[4Ne me(1-nC)]) *NIntegrate[2n(1-n) n^(4Ne me nC-1) (1-n)^(4Ne me(1-nC)-1),{n,0,1}];*)
H=2*4Ne me nC(1-nC)/(1+4Ne me);
Return[If[xA==xC||xB==xC,Null,Chop[H]]]
]
```

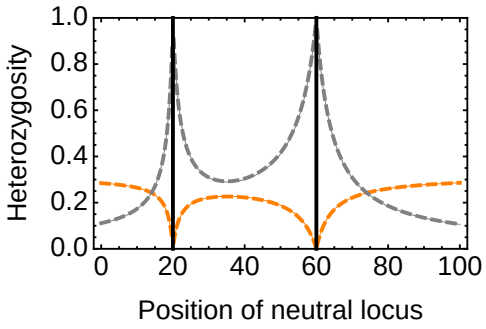
```
Manipulate[Plot[statDistFunc[mya, myb, mym, mynC, myNe, n, myxA, myxB, myxC, mysf],
  {n, 0, 1}, Frame → True, FrameStyle → {{Black, Opacity[0]}, {Black, Opacity[0]}},
  LabelStyle → {Directive[FontSize → 14], FontFamily → "Helvetica"},
  FrameLabel → {"Frequency n of C1", "Density"}, PlotStyle → {Thick, Black}],
 {{mya, 0.2}, 0, 10}, {{myb, 0.8}, 0, 10}, {{mym, 0.24}, 0, 10},
 {{mynC, 0.2}, 0, 1}, {{myNe, 100}, 0, 1000}, {{myxA, 40}, 0, 100},
 {{myxB, 60}, 0, 100}, {{myxC, 50}, 0, 100}, {{mysf, 0.01}, 0, 10}]
```

```
Manipulate[
 Plot3D[statDistFunc[mya, myb, mym, mynC, myNe, n, myxA, myxB, xC, mysf], {n, 0, 1},
  {xC, 0, 100}, LabelStyle → {Directive[FontSize → 14], FontFamily → "Helvetica"},
  AxesLabel → {n, xc}, PlotRange → Automatic, ClippingStyle → None(*,
  ColorFunction→"Rainbow",PlotStyle→Directive[Opacity[0.6]]*), Mesh → None,
  PerformanceGoal → "Quality", PlotPoints → 5], {{mya, 0.2}, 0, 10},
 {{myb, 0.8}, 0, 10}, {{mym, 0.24}, 0, 10}, {{mynC, 0.2}, 0, 1}, {{myNe, 100}, 0, 1000},
 {{myxA, 40}, 0, 100}, {{myxB, 60}, 0, 100}, {{mysf, 0.01}, 0, 10}]
```

```
mya = 0.02;
myb = 0.4;
mym = 0.024;
mynC = 0.4;
myNe = 100;
myxA = 20;
myxB = 60;
mysf = 0.01;
Plot[{effMigRateExactFunc[mya, myb, mym, myNe, myxA, myxB, xC, mysf],
  effMigRateFunc[mya, myb, mym, myNe, myxA, myxB, xC, mysf]}, {xC, 0, 100},
 PlotRange → {Automatic, Automatic}, PlotStyle → {{Red}, {Blue, Thick, Dashed}},
 Frame → True, FrameStyle → {{Black, Black}, {Black, Black}},
 LabelStyle → {Directive[FontSize → 15], FontFamily → "Helvetica"},
 FrameLabel → {"Position of neutral locus", "Effective mig. rate"},
 PlotRange → Automatic, Epilog → {{Black, Thick, Line[{{myxA, 0}, {myxA, 1}}]},
   {Black, Thick, Line[{{myxB, 0}, {myxB, 1}}]},
   {Black, Thick, Dashed, Line[{{0, mynC}, {100, mynC}}]}},
 (*AspectRatio→1,*)ImageSize → 2.5 {100, 80}]
```



```
mya = 0.2;
myb = 4;
mym = 0.24;
mynC = 0.2;
myNe = 100;
myxA = 20;
myxB = 60;
mysf = 0.01;
myRatio = .01;(* This is very low and just for the purpose of illustration.*)
plotFstComb1 = Plot[{hetExactFunc[mya, myb, mym, mynC, myNe, myxA, myxB, xC, mysf],
    hetFuncNew[mya, myb, mym, mynC, myNe, myxA, myxB, xC, mysf, myRatio],
    FstExactFunc[mya, myb, mym, mynC, myNe, myxA, myxB, xC, mysf],
    FstFunc[mya, myb, mym, mynC, myNe, myxA, myxB, xC, mysf, myRatio]},
   {xC, 0, 100}, PlotRange → {Automatic, {0, 1}},
   PlotStyle → {{Orange}, {Orange, Thick, Dashed}, {Gray}, {Gray, Thick, Dashed}},
   Frame → True, FrameStyle → Directive[Thickness[0.005]],
   LabelStyle → {Directive[FontSize → 14], FontFamily → "Helvetica"},
   FrameLabel → {"Position of neutral locus", "Heterozygosity"},
   PlotRange → Automatic, Epilog → {{Black, Thick, Line[{{myxA, 0}, {myxA, 1}}]},
     {Black, Thick, Line[{{myxB, 0}, {myxB, 1}}]}(*,
     {Black,Thick,Dashed,Line[{{0,mynC},{100,mynC}}]}*)},
   (*AspectRatio→1,*)ImageSize → 2.5 {100, 80}]
```




```
Export[PlotPath <> "statDist_hetAndFst_comb1.eps",
 plotFstComb1, "EPS", ImageResolution → 200]
```

```
/Users/Simon/Documents/LocAdD/results/130419/linkedNeutrSite/figures/
  statDist_hetAndFst_comb1.eps
```

```
mya = 0.02;(*0.2;*)
myb = 0.4;(*4;*)
mym = 0.01;(*0.24;*)
mynC = 0.2;
myNe = 100;
myxA = 20;
myxB = 60;
mysf = 0.01;
myRatio = 1;
nPoints = 40;
thicknessSolid = 0.0075;
thicknessDashed = 0.015;
plotFstComb1 = Plot[{hetExactFunc[mya, myb, mym, mynC, myNe, myxA, myxB, xC, mysf],
    hetFunc[mya, myb, mym, mynC, myNe, myxA, myxB, xC, mysf, myRatio],
    FstExactFunc[mya, myb, mym, mynC, myNe, myxA, myxB, xC, mysf],
    FstFunc[mya, myb, mym, mynC, myNe, myxA, myxB, xC, mysf, myRatio]},
  {xC, 0, 100}, PlotRange → {Automatic, Full}, PlotStyle →
   {{Orange, Thickness[thicknessSolid]}, {Orange, Thickness[thicknessDashed], Dashed},
    {White, Thickness[thicknessSolid]}, {White, Thickness[thicknessDashed], Dashed}},
  Frame → True, FrameStyle → Directive[Thickness[0.005]],
  LabelStyle → {Directive[FontSize → 14], FontFamily → "Helvetica"},
  FrameLabel → {"Position of neutral locus", "Heterozygosity"},
  PlotRange → Automatic, Epilog → {{Black, Thick, Line[{{myxA, 0}, {myxA, 1}}]},
    {Black, Thick, Line[{{myxB, 0}, {myxB, 1}}]}(*,
    {Black,Thick,Dashed,Line[{{0,mynC},{100,mynC}}]}*)},
  (*AspectRatio→1,*)ImageSize → 2.5 {100, 80}]; contPlotComb1 =
 ContourPlot[statDistFunc[mya, myb, mym, mynC, myNe, n, myxA, myxB, xC, mysf],
  {xC, 0, 100}, {n, 0, 1}, Frame → True, FrameStyle → Directive[Thickness[0.005]],
  FrameTicks → {{All, All}, {All, None}},
  LabelStyle → {Directive[FontSize → 14], FontFamily → "Helvetica"}(*,FrameLabel→
   {"Position of neutral locus","Frequency n of C1","","Heterozygosity and FST"}*),
  PlotRange → {0, 3}, ClippingStyle → None, PerformanceGoal → "Quality",
  PlotPoints → nPoints, Mesh → {{0}, {myxA, myxB}}, MeshStyle → {Black, Thick},
  Epilog → {{Black, Thick, Dashed, Line[{{myxA, 0}, {myxA, 1}}]}, {Black, Thick, Dashed,
     Line[{{myxB, 0}, {myxB, 1}}]}, {Black, Line[{{0, mynC}, {100, mynC}}]}},
  ColorFunction → ColorData["Gradients"][[14]],
  Contours → Function[{min, max}, Range[min, max, 0.1]](*,
  FrameTicksStyle→Directive[Opacity[1],Thickness[0.0025]]*)];
statDistPlotComb1 = Graphics[{Inset[Show[contPlotComb1, plotFstComb1],
    {0, 0}, {Center, Center}, {100, 100}]}, ImageSize → 2.5 {100, 100}]
```

```
Export[PlotPath <> "statDist_contPlot_comb1a.eps",
 statDistPlotComb1, "EPS", ImageResolution → 300]
```

```
/Users/Simon/Documents/LocAdD/results/130419/linkedNeutrSite/figures/
  statDist_contPlot_comb1a.eps
```

```
mya = 0.02;(*0.2;*)
myb = 0.4;(*4;*)
mym = 0.01;(*0.24;*)
mynC = 0.5;
myNe = 100;
myxA = 20;
myxB = 60;
mysf = 0.01;
myRatio = 1;
nPoints = 40;
thicknessSolid = 0.0075;
thicknessDashed = 0.015;
plotFstComb2 = Plot[{hetExactFunc[mya, myb, mym, mynC, myNe, myxA, myxB, xC, mysf],
    hetFunc[mya, myb, mym, mynC, myNe, myxA, myxB, xC, mysf, myRatio],
    FstExactFunc[mya, myb, mym, mynC, myNe, myxA, myxB, xC, mysf],
    FstFunc[mya, myb, mym, mynC, myNe, myxA, myxB, xC, mysf, myRatio]},
  {xC, 0, 100}, PlotRange → {Automatic, Full}, PlotStyle →
   {{Orange, Thickness[thicknessSolid]}, {Orange, Thickness[thicknessDashed], Dashed},
    {White, Thickness[thicknessSolid]}, {White, Thickness[thicknessDashed], Dashed}},
  Frame → True, FrameStyle → Directive[Thickness[0.005]],
  LabelStyle → {Directive[FontSize → 14], FontFamily → "Helvetica"},
  FrameLabel → {"Position of neutral locus", "Heterozygosity"},
  PlotRange → Automatic, Epilog → {{Black, Thick, Line[{{myxA, 0}, {myxA, 1}}]},
    {Black, Thick, Line[{{myxB, 0}, {myxB, 1}}]}(*,
    {Black,Thick,Dashed,Line[{{0,mynC},{100,mynC}}]}*)},
  (*AspectRatio→1,*)ImageSize → 2.5 {100, 80}]; contPlotComb2 =
 ContourPlot[statDistFunc[mya, myb, mym, mynC, myNe, n, myxA, myxB, xC, mysf],
  {xC, 0, 100}, {n, 0, 1}, Frame → True, FrameStyle → Directive[Thickness[0.005]],
  FrameTicks → {{All, All}, {All, None}},
  LabelStyle → {Directive[FontSize → 14], FontFamily → "Helvetica"}(*,FrameLabel→
   {"Position of neutral locus","Frequency n of C1","","Heterozygosity and FST"}*),
  PlotRange → {0, 2.5}, PerformanceGoal → "Quality", PlotPoints → nPoints,
  Mesh → {{0}, {myxA, myxB}}, MeshStyle → {Black, Thick},
  Epilog → {{Black, Thick, Dashed, Line[{{myxA, 0}, {myxA, 1}}]}, {Black, Thick, Dashed,
     Line[{{myxB, 0}, {myxB, 1}}]}, {Black, Line[{{0, mynC}, {100, mynC}}]}},
  ColorFunction → ColorData["Gradients"][[14]],
  Contours → Function[{min, max}, Range[min, max, 0.1]](*,
  FrameTicksStyle→Directive[Opacity[1],Thickness[0.0025]]*)];
statDistPlotComb2 = Graphics[{Inset[Show[contPlotComb2, plotFstComb2],
    {0, 0}, {Center, Center}, {100, 100}]}, ImageSize → 2.5 {100, 100}]
```

```
Export[PlotPath <> "statDist_contPlot_comb2a.eps",
 statDistPlotComb2, "EPS", ImageResolution → 300]
```

```
/Users/Simon/Documents/LocAdD/results/130419/linkedNeutrSite/figures/
  statDist_contPlot_comb2a.eps
```

```
mya = 0.02;(*0.2;*)
myb = 0.4;(*4;*)
mym = 0.01;(*0.24;*)
mynC = 0.8;
myNe = 100;
myxA = 20;
myxB = 60;
mysf = 0.01;
myRatio = 1;
nPoints = 40;
thicknessSolid = 0.0075;
thicknessDashed = 0.015;
plotFstComb3 = Plot[{hetExactFunc[mya, myb, mym, mynC, myNe, myxA, myxB, xC, mysf],
    hetFunc[mya, myb, mym, mynC, myNe, myxA, myxB, xC, mysf, myRatio],
    FstExactFunc[mya, myb, mym, mynC, myNe, myxA, myxB, xC, mysf],
    FstFunc[mya, myb, mym, mynC, myNe, myxA, myxB, xC, mysf, myRatio]},
  {xC, 0, 100}, PlotRange → {Automatic, Full}, PlotStyle →
   {{Orange, Thickness[thicknessSolid]}, {Orange, Thickness[thicknessDashed], Dashed},
    {White, Thickness[thicknessSolid]}, {White, Thickness[thicknessDashed], Dashed}},
  Frame → True, FrameStyle → Directive[Thickness[0.005]],
  LabelStyle → {Directive[FontSize → 14], FontFamily → "Helvetica"},
  FrameLabel → {"Position of neutral locus", "Heterozygosity"},
  PlotRange → Automatic, Epilog → {{Black, Thick, Line[{{myxA, 0}, {myxA, 1}}]},
    {Black, Thick, Line[{{myxB, 0}, {myxB, 1}}]}(*,
    {Black,Thick,Dashed,Line[{{0,mynC},{100,mynC}}]}*)},
  (*AspectRatio→1,*)ImageSize → 2.5 {100, 80}]; contPlotComb3 =
 ContourPlot[statDistFunc[mya, myb, mym, mynC, myNe, n, myxA, myxB, xC, mysf],
  {xC, 0, 100}, {n, 0, 1}, Frame → True, FrameStyle → Directive[Thickness[0.005]],
  FrameTicks → {{All, All}, {All, None}},
  LabelStyle → {Directive[FontSize → 14], FontFamily → "Helvetica"}(*,FrameLabel→
   {"Position of neutral locus","Frequency n of C1","","Heterozygosity and FST"}*),
  PlotRange → {0, 3}, PerformanceGoal → "Quality", PlotPoints → nPoints,
  Mesh → {{0}, {myxA, myxB}}, MeshStyle → {Black, Thick},
  Epilog → {{Black, Thick, Dashed, Line[{{myxA, 0}, {myxA, 1}}]}, {Black, Thick, Dashed,
     Line[{{myxB, 0}, {myxB, 1}}]}, {Black, Line[{{0, mynC}, {100, mynC}}]}},
  ColorFunction → ColorData["Gradients"][[14]],
  Contours → Function[{min, max}, Range[min, max, 0.1]](*,
  FrameTicksStyle→Directive[Opacity[1],Thickness[0.0025]]*)];
statDistPlotComb3 = Graphics[{Inset[Show[contPlotComb3, plotFstComb3],
    {0, 0}, {Center, Center}, {100, 100}]}, ImageSize → 2.5 {100, 100}]
```

```
Export[PlotPath <> "statDist_contPlot_comb3a.eps",
 statDistPlotComb3, "EPS", ImageResolution → 300]
```

```
/Users/Simon/Documents/LocAdD/results/130419/linkedNeutrSite/figures/
  statDist_contPlot_comb3a.eps
```

```
mya = 0.2;(*0.02;0.2;*)
myb = 4;(*0.04;4;*)
mym = 0.24;(*0.024;0.01;0.24;*)
mynC = 0.2;
myNe = 100;
myxA = 60;
myxB = 100;
mysf = 0.01;
myRatio = 0.01;
nPoints = 40;
thicknessSolid = 0.0075;
thicknessDashed = 0.015;
plotFstComb4 = Plot[{hetExactFunc[mya, myb, mym, mynC, myNe, myxA, myxB, xC, mysf],
    hetFunc[mya, myb, mym, mynC, myNe, myxA, myxB, xC, mysf, myRatio],
    FstExactFunc[mya, myb, mym, mynC, myNe, myxA, myxB, xC, mysf],
    FstFunc[mya, myb, mym, mynC, myNe, myxA, myxB, xC, mysf, myRatio]},
  {xC, 0, 200}, PlotRange → {Automatic, Full}, PlotStyle →
   {{Orange, Thickness[thicknessSolid]}, {Orange, Thickness[thicknessDashed], Dashed},
    {White, Thickness[thicknessSolid]}, {White, Thickness[thicknessDashed], Dashed}},
  Frame → True, FrameStyle → Directive[Thickness[0.005]],
  LabelStyle → {Directive[FontSize → 14], FontFamily → "Helvetica"},
  FrameLabel → {"Position of neutral locus", "Heterozygosity"},
  PlotRange → Automatic, Epilog → {{Black, Thick, Line[{{myxA, 0}, {myxA, 1}}]},
    {Black, Thick, Line[{{myxB, 0}, {myxB, 1}}]}(*,
    {Black,Thick,Dashed,Line[{{0,mynC},{100,mynC}}]}*)},
  (*AspectRatio→1,*)ImageSize → 2.5 {100, 80}]; contPlotComb4 =
 ContourPlot[statDistFunc[mya, myb, mym, mynC, myNe, n, myxA, myxB, xC, mysf],
  {xC, 0, 200}, {n, 0, 1}, Frame → True, FrameStyle → Directive[Thickness[0.005]],
  FrameTicks → {{All, All}, {All, None}},
  LabelStyle → {Directive[FontSize → 14], FontFamily → "Helvetica"}(*,FrameLabel→
   {"Position of neutral locus","Frequency n of C1","","Heterozygosity and FST"}*),
  PlotRange → {0, 5}, PerformanceGoal → "Quality", PlotPoints → nPoints,
  Mesh → {{0}, {myxA, myxB}}, MeshStyle → {Black, Thick},
  Epilog → {{Black, Thick, Dashed, Line[{{myxA, 0}, {myxA, 1}}]}, {Black, Thick, Dashed,
     Line[{{myxB, 0}, {myxB, 1}}]}, {Black, Line[{{0, mynC}, {200, mynC}}]}},
  ColorFunction → ColorData["Gradients"][[14]],
  Contours → Function[{min, max}, Range[min, max, 0.1]](*,
  FrameTicksStyle→Directive[Opacity[1],Thickness[0.0025]]*)];
statDistPlotComb4 = Graphics[{Inset[Show[contPlotComb4, plotFstComb4],
    {0, 0}, {Center, Center}, {200, 200}]}, ImageSize → 2.5 {100, 100}]
```

```
mya = 0.2;(*0.02;0.2;*)
myb = 4;(*0.04;4;*)
mym = 0.24;(*0.024;0.01;0.24;*)
mynC = 0.2;
myNe = 100;
myxA = 20;
myxB = 60;
mysf = 0.01;
myRatio = 0.01;
nPoints = 40;
thicknessSolid = 0.0075;
thicknessDashed = 0.015;
plotFstComb4 = Plot[{hetExactFunc[mya, myb, mym, mynC, myNe, myxA, myxB, xC, mysf],
    hetFunc[mya, myb, mym, mynC, myNe, myxA, myxB, xC, mysf, myRatio],
    FstExactFunc[mya, myb, mym, mynC, myNe, myxA, myxB, xC, mysf],
    FstFunc[mya, myb, mym, mynC, myNe, myxA, myxB, xC, mysf, myRatio]},
  {xC, 0, 100}, PlotRange → {Automatic, Full}, PlotStyle →
   {{Orange, Thickness[thicknessSolid]}, {Orange, Thickness[thicknessDashed], Dashed},
    {White, Thickness[thicknessSolid]}, {White, Thickness[thicknessDashed], Dashed}},
  Frame → True, FrameStyle → Directive[Thickness[0.005]],
  LabelStyle → {Directive[FontSize → 14], FontFamily → "Helvetica"},
  FrameLabel → {"Position of neutral locus", "Heterozygosity"},
  PlotRange → Automatic, Epilog → {{Black, Thick, Line[{{myxA, 0}, {myxA, 1}}]},
    {Black, Thick, Line[{{myxB, 0}, {myxB, 1}}]}(*,
    {Black,Thick,Dashed,Line[{{0,mynC},{100,mynC}}]}*)},
  (*AspectRatio→1,*)ImageSize → 2.5 {100, 80}]; contPlotComb4 =
 ContourPlot[statDistFunc[mya, myb, mym, mynC, myNe, n, myxA, myxB, xC, mysf],
  {xC, 0, 100}, {n, 0, 1}, Frame → True, FrameStyle → Directive[Thickness[0.005]],
  FrameTicks → {{All, All}, {All, None}},
  LabelStyle → {Directive[FontSize → 14], FontFamily → "Helvetica"}(*,FrameLabel→
   {"Position of neutral locus","Frequency n of C1","","Heterozygosity and FST"}*),
  PlotRange → {0, 5}, PerformanceGoal → "Quality", PlotPoints → nPoints,
  Mesh → {{0}, {myxA, myxB}}, MeshStyle → {Black, Thick},
  Epilog → {{Black, Thick, Dashed, Line[{{myxA, 0}, {myxA, 1}}]}, {Black, Thick, Dashed,
     Line[{{myxB, 0}, {myxB, 1}}]}, {Black, Line[{{0, mynC}, {100, mynC}}]}},
  ColorFunction → ColorData["Gradients"][[14]],
  Contours → Function[{min, max}, Range[min, max, 0.1]](*,
  FrameTicksStyle→Directive[Opacity[1],Thickness[0.0025]]*)];
statDistPlotComb4 = Graphics[{Inset[Show[contPlotComb4, plotFstComb4],
    {0, 0}, {Center, Center}, {100, 100}]}, ImageSize → 2.5 {100, 100}]
```

```
Export[PlotPath <> "statDist_contPlot_comb4a.eps",
 statDistPlotComb4, "EPS", ImageResolution → 300]
```

```
/Users/Simon/Documents/LocAdD/results/130419/linkedNeutrSite/figures/
  statDist_contPlot_comb4a.eps
```

```
mya = 0.2;(*0.02;0.2;*)
myb = 4;(*0.04;4;*)
mym = 0.24;(*0.024;0.01;0.24;*)
mynC = 0.5;
myNe = 100;
myxA = 20;
myxB = 60;
mysf = 0.01;
myRatio = 0.01;
thicknessSolid = 0.0075;
thicknessDashed = 0.015;
plotFstComb5 = Plot[{hetExactFunc[mya, myb, mym, mynC, myNe, myxA, myxB, xC, mysf],
    hetFunc[mya, myb, mym, mynC, myNe, myxA, myxB, xC, mysf, myRatio],
    FstExactFunc[mya, myb, mym, mynC, myNe, myxA, myxB, xC, mysf],
    FstFunc[mya, myb, mym, mynC, myNe, myxA, myxB, xC, mysf, myRatio]},
  {xC, 0, 100}, PlotRange → {Automatic, Full}, PlotStyle →
    {{Orange, Thickness[thicknessSolid]}, {Orange, Thickness[thicknessDashed], Dashed},
     {White, Thickness[thicknessSolid]}, {White, Thickness[thicknessDashed], Dashed}},
  Frame → True, FrameStyle → Directive[Thickness[0.005]],
  LabelStyle → {Directive[FontSize → 14], FontFamily → "Helvetica"},
  FrameLabel → {"Position of neutral locus", "Heterozygosity"},
  PlotRange → Automatic, Epilog → {{Black, Thick, Line[{{myxA, 0}, {myxA, 1}}]},
    {Black, Thick, Line[{{myxB, 0}, {myxB, 1}}]}(*,
    {Black,Thick,Dashed,Line[{{0,mynC},{100,mynC}}]}*)},
  (*AspectRatio→1,*)ImageSize → 2.5 {100, 80}]; contPlotComb5 =
 ContourPlot[statDistFunc[mya, myb, mym, mynC, myNe, n, myxA, myxB, xC, mysf],
  {xC, 0, 100}, {n, 0, 1}, Frame → True, FrameStyle → Directive[Thickness[0.005]],
  FrameTicks → {{All, All}, {All, None}},
  LabelStyle → {Directive[FontSize → 14], FontFamily → "Helvetica"}(*,FrameLabel→
   {"Position of neutral locus","Frequency n of C1","","Heterozygosity and FST"}*),
  PlotRange → {0, 3}, PerformanceGoal → "Quality", PlotPoints → nPoints,
  Mesh → {{0}, {myxA, myxB}}, MeshStyle → {Black, Thick},
  Epilog → {{Black, Thick, Dashed, Line[{{myxA, 0}, {myxA, 1}}]}, {Black, Thick, Dashed,
     Line[{{myxB, 0}, {myxB, 1}}]}, {Black, Line[{{0, mynC}, {100, mynC}}]}},
  ColorFunction → ColorData["Gradients"][[14]],
  Contours → Function[{min, max}, Range[min, max, 0.1]](*,
  FrameTicksStyle→Directive[Opacity[1],Thickness[0.0025]]*)];
statDistPlotComb5 = Graphics[{Inset[Show[contPlotComb5, plotFstComb5],
    {0, 0}, {Center, Center}, {100, 100}]}, ImageSize → 2.5 {100, 100}]
```

```
Export[PlotPath <> "statDist_contPlot_comb5a.eps",
 statDistPlotComb5, "EPS", ImageResolution → 300]
```

```
/Users/Simon/Documents/LocAdD/results/130419/linkedNeutrSite/figures/
  statDist_contPlot_comb5a.eps
```

```
mya = 0.2;(*0.02;0.2;*)
myb = 4;(*0.04;4;*)
mym = 0.24;(*0.024;0.01;0.24;*)
mynC = 0.8;
myNe = 100;
myxA = 20;
myxB = 60;
mysf = 0.01;
myRatio = 0.01;
nPoints = 40;
thicknessSolid = 0.0075;
thicknessDashed = 0.015;
plotFstComb6 = Plot[{hetExactFunc[mya, myb, mym, mynC, myNe, myxA, myxB, xC, mysf],
    hetFunc[mya, myb, mym, mynC, myNe, myxA, myxB, xC, mysf, myRatio],
    FstExactFunc[mya, myb, mym, mynC, myNe, myxA, myxB, xC, mysf],
    FstFunc[mya, myb, mym, mynC, myNe, myxA, myxB, xC, mysf, myRatio]},
  {xC, 0, 100}, PlotRange → {Automatic, Full}, PlotStyle →
   {{Orange, Thickness[thicknessSolid]}, {Orange, Thickness[thicknessDashed], Dashed},
    {White, Thickness[thicknessSolid]}, {White, Thickness[thicknessDashed], Dashed}},
  Frame → True, FrameStyle → Directive[Thickness[0.005]],
  LabelStyle → {Directive[FontSize → 14], FontFamily → "Helvetica"},
  FrameLabel → {"Position of neutral locus", "Heterozygosity"},
  PlotRange → Automatic, Epilog → {{Black, Thick, Line[{{myxA, 0}, {myxA, 1}}]},
    {Black, Thick, Line[{{myxB, 0}, {myxB, 1}}]}(*,
    {Black,Thick,Dashed,Line[{{0,mynC},{100,mynC}}]}*)},
  (*AspectRatio→1,*)ImageSize → 2.5 {100, 80}]; contPlotComb6 =
 ContourPlot[statDistFunc[mya, myb, mym, mynC, myNe, n, myxA, myxB, xC, mysf],
  {xC, 0, 100}, {n, 0, 1}, Frame → True, FrameStyle → Directive[Thickness[0.005]],
  LabelStyle → {Directive[FontSize → 14], FontFamily → "Helvetica"},
  FrameTicks → {{All, All}, {All, None}}(*,FrameLabel→
   {"Position of neutral locus","Frequency n of C1","","Heterozygosity and FST"}*),
  PlotRange → {0, 5}, PerformanceGoal → "Quality", PlotPoints → nPoints,
  Mesh → {{0}, {myxA, myxB}}, MeshStyle → {Black, Thick},
  Epilog → {{Black, Thick, Dashed, Line[{{myxA, 0}, {myxA, 1}}]}, {Black, Thick, Dashed,
     Line[{{myxB, 0}, {myxB, 1}}]}, {Black, Line[{{0, mynC}, {100, mynC}}]}},
  ColorFunction → ColorData["Gradients"][[14]],
  Contours → Function[{min, max}, Range[min, max, 0.1]](*,
  FrameTicksStyle→Directive[Opacity[1],Thickness[0.0025]]*)];
statDistPlotComb6 = Graphics[{Inset[Show[contPlotComb6, plotFstComb6],
    {0, 0}, {Center, Center}, {100, 100}]}, ImageSize → 2.5 {100, 100}]
```

```
Export[PlotPath <> "statDist_contPlot_comb6a.eps",
 statDistPlotComb6, "EPS", ImageResolution → 300]
```

```
/Users/Simon/Documents/LocAdD/results/130419/linkedNeutrSite/figures/
  statDist_contPlot_comb6a.eps
```

```
statDistPlotCollection1 =
 Labeled[GraphicsGrid[Partition[MapThread[Labeled[#1, #2, {{Top, Left}},
       LabelStyle → {Directive[FontSize → 20, Bold], FontFamily → "Helvetica"},
       FrameMargins → {{12, 4}, {-2, -10}}] &,
     {{statDistPlotComb1, statDistPlotComb2, statDistPlotComb3, statDistPlotComb4,
       statDistPlotComb5, statDistPlotComb6}, CharacterRange["A", "F"]}], 3],
   ImageSize → 1.4 {620, 390}], {"Position of neutral locus",
   "Frequency n of C1", "Heterozygosity and FST"},
  {Bottom, Left, Right}, RotateLabel → True,
  LabelStyle → {Directive[FontSize → 22], FontFamily → "Helvetica"},
  FrameMargins → {{-6, -6}, {0, 0}}]
```

```
Export[PlotPath <> "statDist_contPlot_collection1.eps",
 statDistPlotCollection1, "EPS", ImageResolution → 300]
```

```
/Users/Simon/Documents/LocAdD/results/130419/linkedNeutrSite/figures/
  statDist_contPlot_collection1.eps
```

Rate of coalescence and coalescence effective size

Basics

We restrict ourselves to the case where migration is strong compared to genetic drift. With this assumption, we may use results by Nagylaki (1980; forward-in-time model) and Notohara (1993; backward-in-time model), described in detail and reviewed by Wakeley (2009).

We assume two demes of size N_1 and N_2 and denote the total number of diploid individuals by $\tilde{N} = N_1 + N_2$. Moreover, we define the relative deme size $c_i = \frac{N_i}{\tilde{N}}$ and let the backward migration rates $m_1 = m_{12}$ and $m_2 = m_{21}$ denote the fractions of individuals in deme 1 and 2 in the current generation that were in deme 2 and 1 in the previous generation, respectively. The strong-migration limit then implies that $N_i\, m_i = \tilde{N}\, c_i\, m_i$ is large. Importantly, the relative deme sizes c_i are constant in the limit of $\tilde{N} \to \infty$.

The whole process of migration and genetic drift is modeled as happening on two time-scales; migration spreads the lineages on a fast time scale, while genetic drift lets lineages coalesce on a slow time scale.

Under these assumptions, it can be shown that the rate of coalescence for a sample of size 2 is independent of whether the two samples were taken from the same or different demes. The rate of coalescence is given by

$$G = \frac{m_2^2}{(m_1 + m_2)^2}\,\frac{1}{c_1} + \frac{m_1^2}{(m_1 + m_2)^2}\,\frac{1}{c_2} \tag{36}$$

(Wakeley 2009, p.193). The rate of coalescence is the weighted average of the equilibrium probabilities that the two lineages were found in the same deme in the previous generation, where the weights are the scaled rates of coalescence in the respective deme (c_i^{-1}).

The coalescence-effective population size is defined as the actual total size multiplied by the inverse of the rate of coalescence, $N_e^{(\text{coal})} = \frac{\tilde{N}}{G}$ (Sj ðin et. al, 2005).

In what follows, we substitute m_1 by the effective migration rate m_e experienced by a neutral locus linked to two sites under selection (*cf.* equations (1) to (3)). To be consistent with the assumption of continent-island migration under which we studied the migration-selection dynamics for the selected loci $\mathcal{A}$ and $\mathcal{B}$, we require $m_1 \gg m_2$. Note that this does not automatically imply $m_e \gg m_2$ though, depending on the selection coefficients a and b and on the recombination rate(s).

In[256]:=
```
G := m2^2/(m1 + m2)^2 1/c1 + m1^2/(m1 + m2)^2 1/c2
```

```
G /. {m1 → 0.0001, m2 → 0.0001, c1 → 0.01, c2 → 0.99}
```

```
25.2525
```

Analytical results

- Generic

```
G /. {m1 → m} /. {ruleMeACB}
```

$$\frac{\text{m2}^2}{\text{c1}\left(\text{m2} + \frac{\text{m rAC rCB}}{(\text{a}+\text{rAC})\,(\text{b}+\text{rCB})}\right)^2} + \frac{\text{m}^2\,\text{rAC}^2\,\text{rCB}^2}{\text{c2}\,(\text{a}+\text{rAC})^2\,(\text{b}+\text{rCB})^2\left(\text{m2} + \frac{\text{m rAC rCB}}{(\text{a}+\text{rAC})\,(\text{b}+\text{rCB})}\right)^2}$$

```
G /. {m1 → m} /. {ruleMeABC}
```

$$\frac{\text{m2}^2}{\text{c1}\left(\text{m2} + \frac{\text{m rBC}\,(\text{b}+\text{rAB}+\text{rBC})}{(\text{b}+\text{rBC})\,(\text{a}+\text{b}+\text{rAB}+\text{rBC})}\right)^2} + \frac{\text{m}^2\,\text{rBC}^2\,(\text{b}+\text{rAB}+\text{rBC})^2}{\text{c2}\,(\text{b}+\text{rBC})^2\,(\text{a}+\text{b}+\text{rAB}+\text{rBC})^2\left(\text{m2} + \frac{\text{m rBC}\,(\text{b}+\text{rAB}+\text{rBC})}{(\text{b}+\text{rBC})\,(\text{a}+\text{b}+\text{rAB}+\text{rBC})}\right)^2}$$

```
G /. {m1 → m} /. {ruleMeCAB}
```

$$\frac{\text{m2}^2}{\text{c1}\left(\text{m2} + \frac{\text{m rCA}\,(\text{a}+\text{rAB}+\text{rCA})}{(\text{a}+\text{rCA})\,(\text{a}+\text{b}+\text{rAB}+\text{rCA})}\right)^2} + \frac{\text{m}^2\,\text{rCA}^2\,(\text{a}+\text{rAB}+\text{rCA})^2}{\text{c2}\,(\text{a}+\text{rCA})^2\,(\text{a}+\text{b}+\text{rAB}+\text{rCA})^2\left(\text{m2} + \frac{\text{m rCA}\,(\text{a}+\text{rAB}+\text{rCA})}{(\text{a}+\text{rCA})\,(\text{a}+\text{b}+\text{rAB}+\text{rCA})}\right)^2}$$

- Assume $m_1 / m_2 = c_2 / c_1$

```
G /. {m2 → m1 c1 / c2} /. {m1 → m} /. {ruleMeACB} // Simplify
```

$$\frac{1}{\text{c1} + \text{c2}}$$

```
G /. {m2 → m1 c1 / c2} /. {m1 → m} /. {ruleMeABC} // Simplify
```

$$\frac{1}{c1 + c2}$$

```
G /. {m2 → m1 c1 / c2} /. {m1 → m} /. {ruleMeCAB} // Simplify
```

$$\frac{1}{c1 + c2}$$

Plots: rate of coalescence and effective size

Preparation

We define a function that computes the rate of coalescence, accounting for linkage to two selected sites.

In[257]:=

```
coalRateFunc::usage =
  "coalRateFunc[a, b, m1, m2, c1, c2, xA, xB, xC, scaleFac, ratio] returns the
    rate of coalescence for a sample of size 2 at a neutral locus linked
    to two sites under selection, valid for scenarios where migration is
    strong compared to genetic drift (Ni m large, where Ni=ci Ñ, Ñ is the
    total number of indivisuals and ci the relative size of deme i), but weak
    compared to selection and recombination (m ≪ min (a, r)). The selection
    coefficients at the two sites are a and b, the actual backward migration
    rates are m1 and m2, and the positions of the three sites in map units
    are xA, xB and xC, where the latter belongs to the neutral locus and
    xA < xB is assumed without loss of generality. The scaleFac defines
    what recombination rate should correspond to one map unit, and ratio
    denotes by how much migration must be weaker compared to recombination.";
coalRateFunc[a_, b_, m1_, m2_, c1_, c2_, xA_, xB_, xC_, scaleFac_, ratio_] :=
 Module[{me, coalRate, rAC, rBC, x},
  rAC = Abs[xA - xC] scaleFac;
  rBC = Abs[xB - xC] scaleFac;
  x = Min[rAC, rBC];
  me = If[xC < xA < xB, m1 rAC (a + rBC) / ((a + rAC) (a + b + rBC)), If[xA ≤ xC ≤ xB,
      m1 rAC rBC / ((a + rAC) (b + rBC)), If[xA < xB < xC, m1 rBC (b + rAC) / ((b + rBC) (a + b + rAC)), Null]]];

  coalRate = m2^2 / (me + m2)^2 1/c1 + me^2 / (me + m2)^2 1/c2;
  Return[If[xA == xC || xB == xC, Null, If[(m1 ratio ≤ x), Chop[coalRate], Null]]]
 ]
```

In[259]:=

```
coalRateExactFunc::usage="coalRateExactFunc[a, b, m1, m2, c1, c2, xA, xB, xC, scaleFac
coalRateExactFunc[a_,b_,m1_,m2_,c1_,c2_,xA_,xB_,xC_,scaleFac_]:=Module[{me,coalRate,rA
rAC=Abs[xA-xC]scaleFac;
rBC=Abs[xB-xC]scaleFac;
JNACB={{-m,a,b,0},{1/(8 a (r1+r2)) m (-a^2+b^2+6 a (r1+r2)-4 m (r1+r2)-(r1+r2)^2+(a-b+r1+r2) √
JNABC={{-m,a,b,0},{m (-a^2+b^2+6 a r1-4 m r1-r1^2+(a-b+r1) √(-8 m r1+(a+b+r1)^2))/(8 a r1),a-m-r1-
JNCAB={{-m,b,a,0},{m (a^2-b^2+6 b r2-4 m r2-r2^2+(-a+b+r2) √(-8 m r2+(a+b+r2)^2))/(8 b r2),b-m-r1-r2
me=If[xC<xA<xB,-Max[Re[Eigenvalues[JNCAB]]],If[xA≤xC≤xB,-Max[Re[Eigenvalues[JNACB]]],I

coalRate=m2^2/(me+m2)^2 1/c1+me^2/(me+m2)^2 1/c2;
Return[If[xA==xC||xB==xC,Null,Chop[coalRate]]]
]
```

```
mya = .1;(* The selection coefficient in favour of A1. *)
myb = .4; (* The selection coefficient in favour of B1. *)
mym1 = 0.01; (* The backward migration rate from deme 1 to deme 2. *)
mym2 = mym1 / 100; (* The backward migration rate from deme 2 to deme 1. *)
myc1 = 0.001; (* The fraction of the population living in deme 1 (island). *)
myc2 = 1 - myc1; (* The fraction of the population living in deme 2 (continent). *)
myxA = 80; (* The posittion in map units of locus A. *)
myxB = 120; (* The posittion in map units of locus B. *)
mysf = 0.01; (* One map unit corresponds to r = mysf. *)
myxC = 20;
coalRateExactFunc[mya, myb, mym1, mym2, myc1, myc2, myxA, myxB, myxC, mysf]
coalRateFunc[mya, myb, mym1, mym2, myc1, myc2, myxA, myxB, myxC, mysf, 0.01]
```

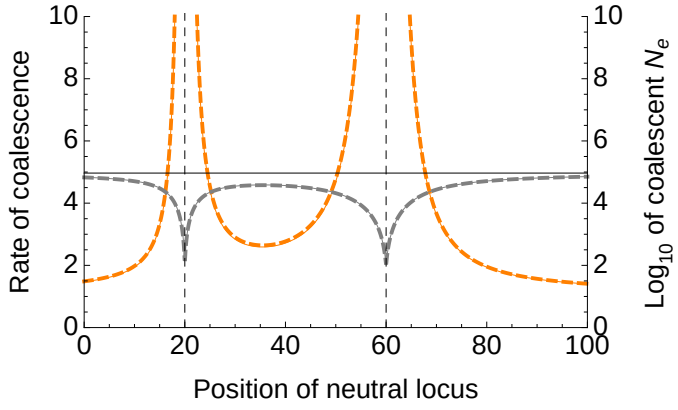
```
1.21088
```

```
1.21513
```

■ Various plots: $\mathcal{A}$ and $\mathcal{B}$ far apart, $c_1 = 0.001$, $m_1 / m_2 = 100$

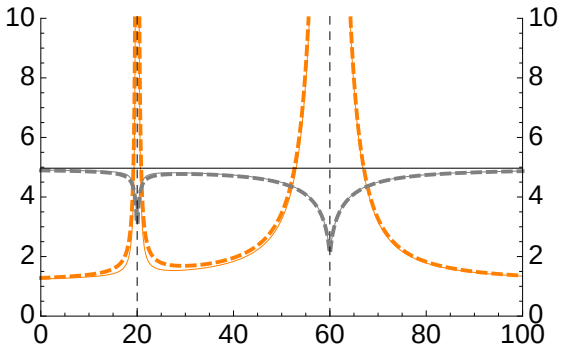
```
mya = .1; (* The selection coefficient in favour of A1. *)
myb = .4; (* The selection coefficient in favour of B1. *)
mym1 = 0.01; (* The backward migration rate from deme 1 to deme 2. *)
mym2 = mym1 / 100; (* The backward migration rate from deme 2 to deme 1. *)
myc1 = 0.001; (* The fraction of the population living in deme 1 (island). *)
myc2 = 1 - myc1; (* The fraction of the population living in deme 2 (continent). *)
myxA = 20; (*80;*) (* The posittion in map units of locus A. *)
myxB = 60; (*120;*) (* The posittion in map units of locus B. *)
mysf = 0.01; (* One map unit corresponds to r = mysf. *)
myRatio = 0.05; (* The minimum factor by which migration
 must be stronger than recombination for points to be plotted. *)
NTilde = 10^8; (* The size of the total population. *)
maxL = 100; (*200;*) (* The length of the chromosome in map units. *)
maxRate = 10;
colNe = GrayLevel[0.5];
plotScaling = 1;
logBase = 10;
MapThread[NTilde * #1 * #2 &, {{myc1, myc2}, {mym1, mym2}}]
plotCoalComb1 =
 Plot[{coalRateExactFunc[mya, myb, mym1, mym2, myc1, myc2, myxA, myxB, xC, mysf],
   coalRateFunc[mya, myb, mym1, mym2, myc1, myc2, myxA, myxB, xC, mysf, myRatio],
   plotScaling * Log[logBase, myc1 *
      NTilde / coalRateExactFunc[mya, myb, mym1, mym2, myc1, myc2, myxA, myxB, xC, mysf]],
   plotScaling * Log[logBase, myc1 * NTilde / coalRateFunc[mya, myb, mym1, mym2, myc1,
       myc2, myxA, myxB, xC, mysf, myRatio]], plotScaling * Log[logBase, myc1 * NTilde /
      coalRateFunc[0, 0, mym1, mym2, myc1, myc2, myxA, myxB, xC, mysf, myRatio]]},
  {xC, 0, maxL}, PlotRange → {{0, maxL}, {0, maxRate}}, Frame → True,
  FrameStyle → {{Black, Black}, {Black, Opacity[0]}},
  FrameTicks → {{All, All}, {All, All}},
  LabelStyle → {Directive[FontSize → 14], FontFamily → "Helvetica"},
  FrameLabel → {{"Rate of coalescence", "Log10 of coalescent Ne"},
    {"Position of neutral locus", ""}},
  PlotStyle → {Orange, {Thick, Orange, Dashed}, colNe, {Thick, colNe, Dashed}, Black},
  Epilog → {{Black, Dashed, Line[{{myxA, 0}, {myxA, 1000}}]},
    {Black, Dashed, Line[{{myxB, 0}, {myxB, 1000}}]}}, ImageSize → 1 {360, 240}]
```

```
{1000., 9990.}
```



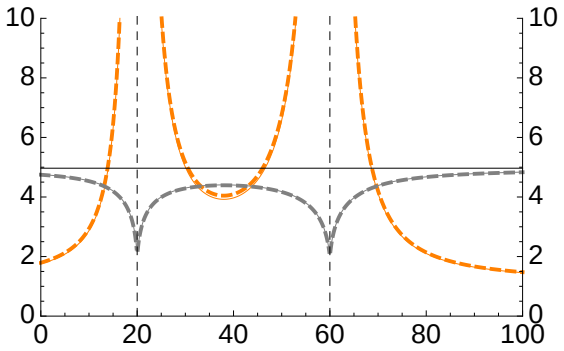
```
mya = .02;(* The selection coefficient in favour of A1. *)
myb = .4; (* The selection coefficient in favour of B1. *)
mym1 = 0.024; (* The backward migration rate from deme 1 to deme 2. *)
mym2 = mym1 / 100; (* The backward migration rate from deme 2 to deme 1. *)
myc1 = 0.001; (* The fraction of the population living in deme 1 (island). *)
myc2 = 1 - myc1; (* The fraction of the population living in deme 2 (continent). *)
myxA = 20;(*80;*) (* The posittion in map units of locus A. *)
myxB = 60;(*120;*) (* The posittion in map units of locus B. *)
mysf = 0.01; (* One map unit corresponds to r = mysf. *)
myRatio = 0.05; (* The minimum factor by which migration
 must be stronger than recombination for points to be plotted. *)
NTilde = 10^8; (* The size of the total population. *)
maxL = 100;(*200;*) (* The length of the chromosome in map units. *)
maxRate = 10;
colNe = GrayLevel[0.5];
plotScaling = 1;
logBase = 10;
MapThread[NTilde * #1 * #2 &, {{myc1, myc2}, {mym1, mym2}}]
plotCoalComb2 =
 Plot[{coalRateExactFunc[mya, myb, mym1, mym2, myc1, myc2, myxA, myxB, xC, mysf],
   coalRateFunc[mya, myb, mym1, mym2, myc1, myc2, myxA, myxB, xC, mysf, myRatio],
   plotScaling * Log[logBase, myc1 *
       NTilde / coalRateExactFunc[mya, myb, mym1, mym2, myc1, myc2, myxA, myxB, xC, mysf]],
   plotScaling * Log[logBase, myc1 * NTilde / coalRateFunc[mya, myb, mym1, mym2, myc1,
         myc2, myxA, myxB, xC, mysf, myRatio]], plotScaling * Log[logBase, myc1 * NTilde /
       coalRateFunc[0, 0, mym1, mym2, myc1, myc2, myxA, myxB, xC, mysf, myRatio]]},
  {xC, 0, maxL}, PlotRange → {{0, maxL}, {0, maxRate}}, Frame → True,
  FrameStyle → {{Black, Black}, {Black, Opacity[0]}},
  FrameTicks → {{All, All}, {All, All}},
  LabelStyle → {Directive[FontSize → 14], FontFamily → "Helvetica"}
  (*,FrameLabel→{{"Rate of coalescence","Log10 of coalescent Ne"},
    {"Position of neutral locus",""}}*),
  PlotStyle → {Orange, {Thick, Orange, Dashed}, colNe, {Thick, colNe, Dashed}, Black},
  Epilog → {{Black, Dashed, Line[{{myxA, 0}, {myxA, 1000}}]},
    {Black, Dashed, Line[{{myxB, 0}, {myxB, 1000}}]}}, ImageSize → 0.8 {360, 240}]
```

```
{2400., 23976.}
```



```
mya = .2;(* The selection coefficient in favour of A1. *)
myb = .4; (* The selection coefficient in favour of B1. *)
mym1 = 0.024; (* The backward migration rate from deme 1 to deme 2. *)
mym2 = mym1 / 100; (* The backward migration rate from deme 2 to deme 1. *)
myc1 = 0.001; (* The fraction of the population living in deme 1 (island). *)
myc2 = 1 - myc1; (* The fraction of the population living in deme 2 (continent). *)
myxA = 20;(*80;*) (* The posittion in map units of locus A. *)
myxB = 60;(*120;*) (* The posittion in map units of locus B. *)
mysf = 0.01; (* One map unit corresponds to r = mysf. *)
myRatio = 0.05; (* The minimum factor by which migration
 must be stronger than recombination for points to be plotted. *)
NTilde = 10^8; (* The size of the total population. *)
maxL = 100;(*200;*) (* The length of the chromosome in map units. *)
maxRate = 10;
colNe = GrayLevel[0.5];
plotScaling = 1;
logBase = 10;
MapThread[NTilde * #1 * #2 &, {{myc1, myc2}, {mym1, mym2}}]
plotCoalComb3 =
 Plot[{coalRateExactFunc[mya, myb, mym1, mym2, myc1, myc2, myxA, myxB, xC, mysf],
   coalRateFunc[mya, myb, mym1, mym2, myc1, myc2, myxA, myxB, xC, mysf, myRatio],
   plotScaling * Log[logBase, myc1 *
      NTilde / coalRateExactFunc[mya, myb, mym1, mym2, myc1, myc2, myxA, myxB, xC, mysf]],
   plotScaling * Log[logBase, myc1 * NTilde / coalRateFunc[mya, myb, mym1, mym2, myc1,
        myc2, myxA, myxB, xC, mysf, myRatio]], plotScaling * Log[logBase, myc1 * NTilde /
       coalRateFunc[0, 0, mym1, mym2, myc1, myc2, myxA, myxB, xC, mysf, myRatio]]},
  {xC, 0, maxL}, PlotRange → {{0, maxL}, {0, maxRate}}, Frame → True,
  FrameStyle → {{Black, Black}, {Black, Opacity[0]}},
  FrameTicks → {{All, All}, {All, All}},
  LabelStyle → {Directive[FontSize → 14], FontFamily → "Helvetica"}
  (*,FrameLabel→{{"Rate of coalescence","Log10 of coalescent Ne"},
    {"Position of neutral locus",""}}*),
  PlotStyle → {Orange, {Thick, Orange, Dashed}, colNe, {Thick, colNe, Dashed}, Black},
  Epilog → {{Black, Dashed, Line[{{myxA, 0}, {myxA, 1000}}]},
    {Black, Dashed, Line[{{myxB, 0}, {myxB, 1000}}]}}, ImageSize → 0.8 {360, 240}]
```

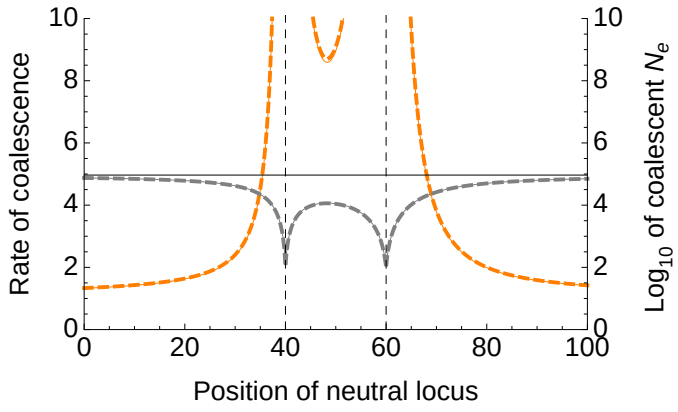
```
{2400., 23976.}
```



■ Various plots: $\mathcal{A}$ and $\mathcal{B}$ rather close, $c_1 = 0.001$, $m_1 / m_2 = 100$

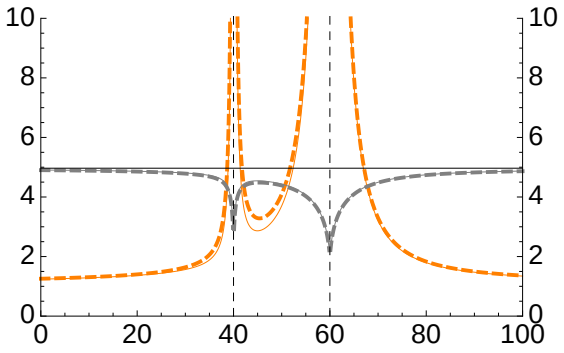
```
mya = .1; (* The selection coefficient in favour of A1. *)
myb = .4; (* The selection coefficient in favour of B1. *)
mym1 = 0.01; (* The backward migration rate from deme 1 to deme 2. *)
mym2 = mym1 / 100; (* The backward migration rate from deme 2 to deme 1. *)
myc1 = 0.001; (* The fraction of the population living in deme 1 (island). *)
myc2 = 1 - myc1; (* The fraction of the population living in deme 2 (continent). *)
myxA = 40; (*80;*) (* The posittion in map units of locus A. *)
myxB = 60; (*120;*) (* The posittion in map units of locus B. *)
mysf = 0.01; (* One map unit corresponds to r = mysf. *)
myRatio = 0.05; (* The minimum factor by which migration
 must be stronger than recombination for points to be plotted. *)
NTilde = 10^8; (* The size of the total population. *)
maxL = 100; (*200;*) (* The length of the chromosome in map units. *)
maxRate = 10;
colNe = GrayLevel[0.5];
plotScaling = 1;
logBase = 10;
MapThread[NTilde * #1 * #2 &, {{myc1, myc2}, {mym1, mym2}}]
plotCoalComb4 =
 Plot[{coalRateExactFunc[mya, myb, mym1, mym2, myc1, myc2, myxA, myxB, xC, mysf],
   coalRateFunc[mya, myb, mym1, mym2, myc1, myc2, myxA, myxB, xC, mysf, myRatio],
   plotScaling * Log[logBase, myc1 *
      NTilde / coalRateExactFunc[mya, myb, mym1, mym2, myc1, myc2, myxA, myxB, xC, mysf]],
   plotScaling * Log[logBase, myc1 * NTilde / coalRateFunc[mya, myb, mym1, mym2, myc1,
       myc2, myxA, myxB, xC, mysf, myRatio]], plotScaling * Log[logBase, myc1 * NTilde /
      coalRateFunc[0, 0, mym1, mym2, myc1, myc2, myxA, myxB, xC, mysf, myRatio]]},
  {xC, 0, maxL}, PlotRange → {{0, maxL}, {0, maxRate}}, Frame → True,
  FrameStyle → {{Black, Black}, {Black, Opacity[0]}},
  FrameTicks → {{All, All}, {All, All}},
  LabelStyle → {Directive[FontSize → 14], FontFamily → "Helvetica"},
  FrameLabel → {{"Rate of coalescence", "Log10 of coalescent Ne "},
    {"Position of neutral locus", ""}},
  PlotStyle → {Orange, {Thick, Orange, Dashed}, colNe, {Thick, colNe, Dashed}, Black},
  Epilog → {{Black, Dashed, Line[{{myxA, 0}, {myxA, 1000}}]},
    {Black, Dashed, Line[{{myxB, 0}, {myxB, 1000}}]}}, ImageSize → 1 {360, 240}]
```

```
{1000., 9990.}
```



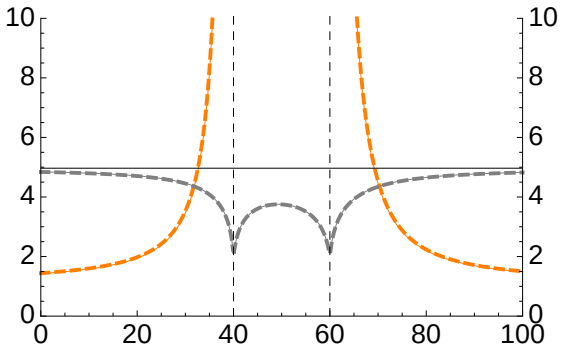
```
mya = .02;(* The selection coefficient in favour of A1. *)
myb = .4; (* The selection coefficient in favour of B1. *)
mym1 = 0.024; (* The backward migration rate from deme 1 to deme 2. *)
mym2 = mym1 / 100; (* The backward migration rate from deme 2 to deme 1. *)
myc1 = 0.001; (* The fraction of the population living in deme 1 (island). *)
myc2 = 1 - myc1; (* The fraction of the population living in deme 2 (continent). *)
myxA = 40;(*80;*) (* The posittion in map units of locus A. *)
myxB = 60;(*120;*) (* The posittion in map units of locus B. *)
mysf = 0.01; (* One map unit corresponds to r = mysf. *)
myRatio = 0.05; (* The minimum factor by which migration
 must be stronger than recombination for points to be plotted. *)
NTilde = 10^8; (* The size of the total population. *)
maxL = 100;(*200;*) (* The length of the chromosome in map units. *)
maxRate = 10;
colNe = GrayLevel[0.5];
plotScaling = 1;
logBase = 10;
MapThread[NTilde * #1 * #2 &, {{myc1, myc2}, {mym1, mym2}}]
plotCoalComb5 =
 Plot[{coalRateExactFunc[mya, myb, mym1, mym2, myc1, myc2, myxA, myxB, xC, mysf],
   coalRateFunc[mya, myb, mym1, mym2, myc1, myc2, myxA, myxB, xC, mysf, myRatio],
   plotScaling * Log[logBase, myc1 *
      NTilde / coalRateExactFunc[mya, myb, mym1, mym2, myc1, myc2, myxA, myxB, xC, mysf]],
   plotScaling * Log[logBase, myc1 * NTilde / coalRateFunc[mya, myb, mym1, mym2, myc1,
        myc2, myxA, myxB, xC, mysf, myRatio]], plotScaling * Log[logBase, myc1 * NTilde /
       coalRateFunc[0, 0, mym1, mym2, myc1, myc2, myxA, myxB, xC, mysf, myRatio]]},
  {xC, 0, maxL}, PlotRange → {{0, maxL}, {0, maxRate}}, Frame → True,
  FrameStyle → {{Black, Black}, {Black, Opacity[0]}},
  FrameTicks → {{All, All}, {All, All}},
  LabelStyle → {Directive[FontSize → 14], FontFamily → "Helvetica"}
  (*,FrameLabel→{{"Rate of coalescence","Log10 of coalescent Ne"},
    {"Position of neutral locus",""}}*),
  PlotStyle → {Orange, {Thick, Orange, Dashed}, colNe, {Thick, colNe, Dashed}, Black},
  Epilog → {{Black, Dashed, Line[{{myxA, 0}, {myxA, 1000}}]},
    {Black, Dashed, Line[{{myxB, 0}, {myxB, 1000}}]}}, ImageSize → 0.8 {360, 240}]
```

```
{2400., 23976.}
```



```
mya = .2;(* The selection coefficient in favour of A1. *)
myb = .4; (* The selection coefficient in favour of B1. *)
mym1 = 0.024; (* The backward migration rate from deme 1 to deme 2. *)
mym2 = mym1 / 100; (* The backward migration rate from deme 2 to deme 1. *)
myc1 = 0.001; (* The fraction of the population living in deme 1 (island). *)
myc2 = 1 - myc1; (* The fraction of the population living in deme 2 (continent). *)
myxA = 40;(*80;*) (* The posittion in map units of locus A. *)
myxB = 60;(*120;*) (* The posittion in map units of locus B. *)
mysf = 0.01; (* One map unit corresponds to r = mysf. *)
myRatio = 0.05; (* The minimum factor by which migration
 must be stronger than recombination for points to be plotted. *)
NTilde = 10^8; (* The size of the total population. *)
maxL = 100;(*200;*) (* The length of the chromosome in map units. *)
maxRate = 10;
colNe = GrayLevel[0.5];
plotScaling = 1;
logBase = 10;
MapThread[NTilde * #1 * #2 &, {{myc1, myc2}, {mym1, mym2}}]
plotCoalComb6 =
 Plot[{coalRateExactFunc[mya, myb, mym1, mym2, myc1, myc2, myxA, myxB, xC, mysf],
   coalRateFunc[mya, myb, mym1, mym2, myc1, myc2, myxA, myxB, xC, mysf, myRatio],
   plotScaling * Log[logBase, myc1 *
      NTilde / coalRateExactFunc[mya, myb, mym1, mym2, myc1, myc2, myxA, myxB, xC, mysf]],
   plotScaling * Log[logBase, myc1 * NTilde / coalRateFunc[mya, myb, mym1, mym2, myc1,
        myc2, myxA, myxB, xC, mysf, myRatio]], plotScaling * Log[logBase, myc1 * NTilde /
       coalRateFunc[0, 0, mym1, mym2, myc1, myc2, myxA, myxB, xC, mysf, myRatio]]},
  {xC, 0, maxL}, PlotRange → {{0, maxL}, {0, maxRate}}, Frame → True,
  FrameStyle → {{Black, Black}, {Black, Opacity[0]}},
  FrameTicks → {{All, All}, {All, All}},
  LabelStyle → {Directive[FontSize → 14], FontFamily → "Helvetica"}
  (*,FrameLabel→{{"Rate of coalescence","Log10 of coalescent Ne"},
    {"Position of neutral locus",""}}*),
  PlotStyle → {Orange, {Thick, Orange, Dashed}, colNe, {Thick, colNe, Dashed}, Black},
  Epilog → {{Black, Dashed, Line[{{myxA, 0}, {myxA, 1000}}]},
    {Black, Dashed, Line[{{myxB, 0}, {myxB, 1000}}]}}, ImageSize → 0.8 {360, 240}]
```

```
{2400., 23976.}
```



```
coalPlot1 =
 Labeled[GraphicsGrid[Partition[MapThread[Labeled[#1, #2, {{Top, Left}}, LabelStyle →
         {Directive[FontSize → 20, Bold], FontFamily → "Helvetica"},
        FrameMargins → {{12, 4}, {-2, -10}}] &, {{plotCoalComb2, plotCoalComb5,
        plotCoalComb3, plotCoalComb6}, CharacterRange["A", "D"]}], 2],
   ImageSize → 1.1 {600, 405}, AspectRatio → 0.7], {"Position of neutral locus",
   "Rate of coalescence", "Log10 of coalescent Ne"},
  {Bottom, Left, Right}, RotateLabel → True,
  LabelStyle → {Directive[FontSize → 20], FontFamily → "Helvetica"},
  FrameMargins → {{0, -2}, {5, 0}}]
```

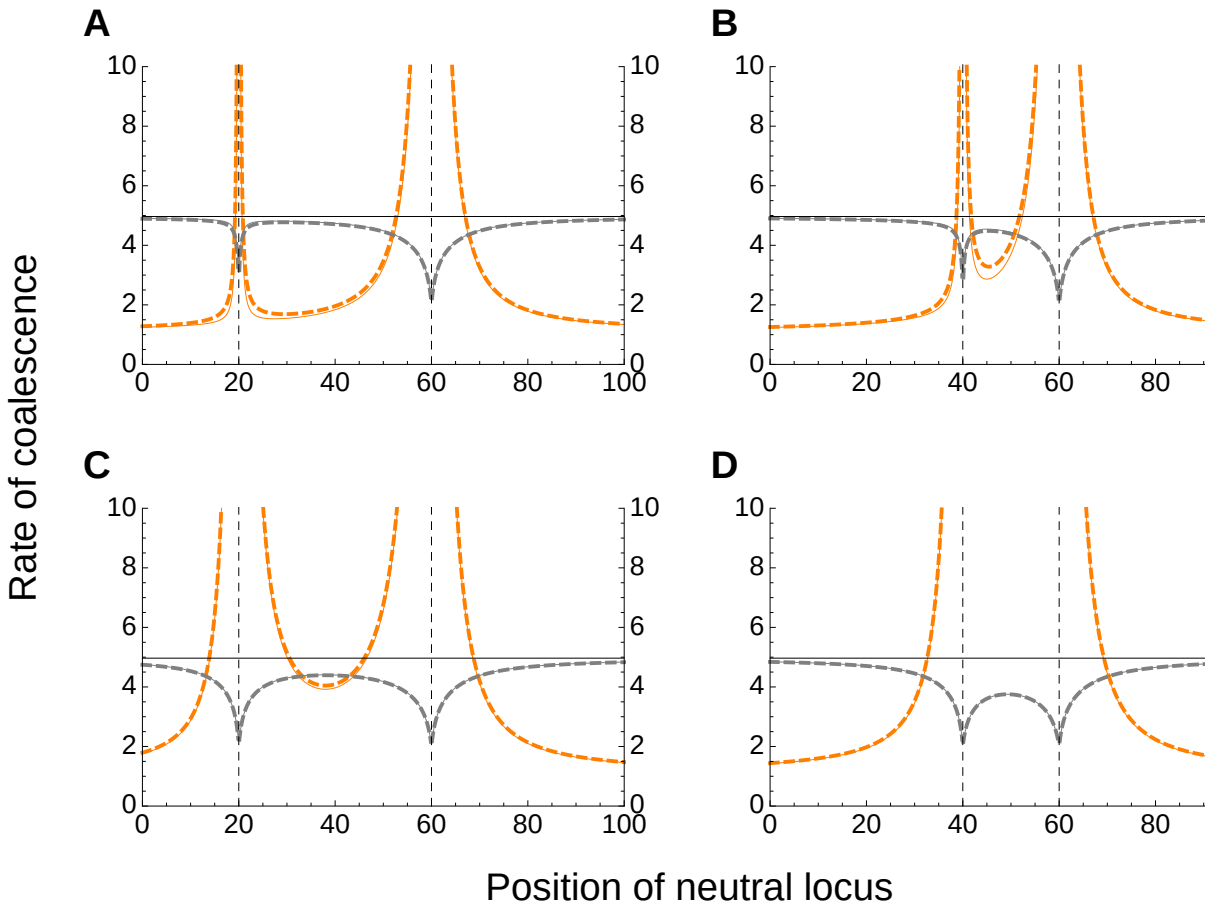


Figure 8: The effect on neutral coalescence of linkage to two sites at migration-selection balance. The rate of coalescence G (orange) and the coalescent effective size $\tilde{N}/G$ (gray) are given as a function of the position (in map units) of the neutral locus. Solid and dashed lines belong to values computed using the exact and approximate (Eq. 1 to 3) effective migration rate, respectively. One map unit (cM) is assumed to correspond to a recombination rate of $r = 0.01$ and the position of the sites under selection is indicated by vertical dashed lines. The total population size is $\tilde{N} = 10^8$, the fraction of the island is $c_1 = 0.01$ and the selection coefficient at locus $\mathcal{B}$ (position 60) is $b = 0.4$. (A) and (B) Migration rate to the island of the same order of magnitude as selection at locus $\mathcal{A}$: $a = 0.02$, $m_1 = 0.024$. (C) and (D) Weak immigration compared to selection at locus $\mathcal{A}$: $a = 0.2$, $m_1 = 0.024$. Throughout, $m_1/m_2 = 100$.

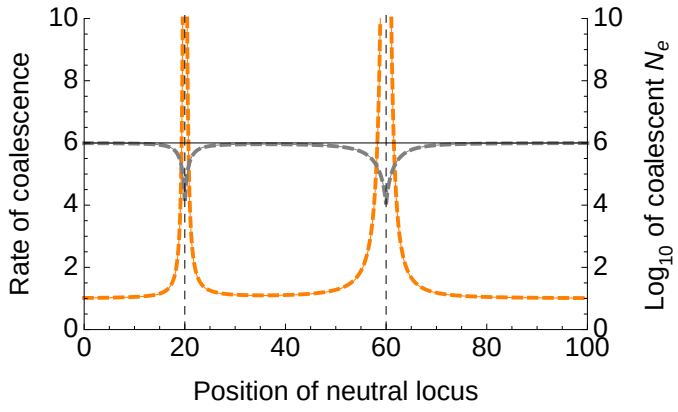
```
Export[PlotPath <> "coalRateAndNe_comb1.eps", coalPlot1, "EPS"]
```

```
/Users/Simon/Documents/LocAdD/results/130419/linkedNeutrSite/figures/
  coalRateAndNe_comb1.eps
```

■ Various plots: $\mathcal{A}$ and $\mathcal{B}$ far apart, $c_1 = 0.01$, $m_1 / m_2 = 100$

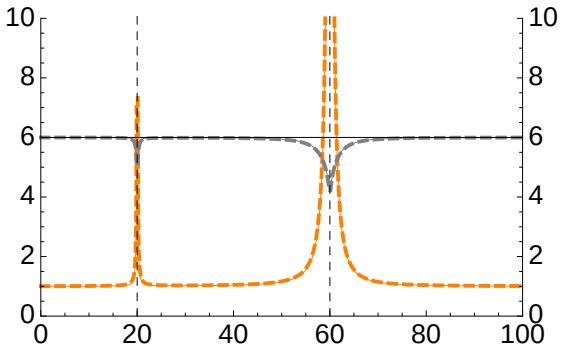
```
mya = .1; (* The selection coefficient in favour of A1. *)
myb = .4; (* The selection coefficient in favour of B1. *)
mym1 = 0.01; (* The backward migration rate from deme 1 to deme 2. *)
mym2 = mym1 / 100; (* The backward migration rate from deme 2 to deme 1. *)
myc1 = 0.01; (* The fraction of the population living in deme 1 (island). *)
myc2 = 1 - myc1; (* The fraction of the population living in deme 2 (continent). *)
myxA = 20; (*80;*) (* The posittion in map units of locus A. *)
myxB = 60; (*120;*) (* The posittion in map units of locus B. *)
mysf = 0.01; (* One map unit corresponds to r = mysf. *)
myRatio = 0.05; (* The minimum factor by which migration
 must be stronger than recombination for points to be plotted. *)
NTilde = 10^8; (* The size of the total population. *)
maxL = 100; (*200;*) (* The length of the chromosome in map units. *)
maxRate = 10;
colNe = GrayLevel[0.5];
plotScaling = 1;
logBase = 10;
MapThread[NTilde * #1 * #2 &, {{myc1, myc2}, {mym1, mym2}}]
plotCoalComb7 =
 Plot[{coalRateExactFunc[mya, myb, mym1, mym2, myc1, myc2, myxA, myxB, xC, mysf],
   coalRateFunc[mya, myb, mym1, mym2, myc1, myc2, myxA, myxB, xC, mysf, myRatio],
   plotScaling * Log[logBase, myc1 *
      NTilde / coalRateExactFunc[mya, myb, mym1, mym2, myc1, myc2, myxA, myxB, xC, mysf]],
   plotScaling * Log[logBase, myc1 * NTilde / coalRateFunc[mya, myb, mym1, mym2, myc1,
       myc2, myxA, myxB, xC, mysf, myRatio]], plotScaling * Log[logBase, myc1 * NTilde /
      coalRateFunc[0, 0, mym1, mym2, myc1, myc2, myxA, myxB, xC, mysf, myRatio]]},
  {xC, 0, maxL}, PlotRange → {{0, maxL}, {0, maxRate}}, Frame → True,
  FrameStyle → {{Black, Black}, {Black, Opacity[0]}},
  FrameTicks → {{All, All}, {All, All}},
  LabelStyle → {Directive[FontSize → 14], FontFamily → "Helvetica"},
  FrameLabel → {{"Rate of coalescence", "Log10 of coalescent Ne"},
    {"Position of neutral locus", ""}},
  PlotStyle → {Orange, {Thick, Orange, Dashed}, colNe, {Thick, colNe, Dashed}, Black},
  Epilog → {{Black, Dashed, Line[{{myxA, 0}, {myxA, 1000}}]},
    {Black, Dashed, Line[{{myxB, 0}, {myxB, 1000}}]}}, ImageSize → 1 {360, 240}]
```

```
{10 000., 9900.}
```



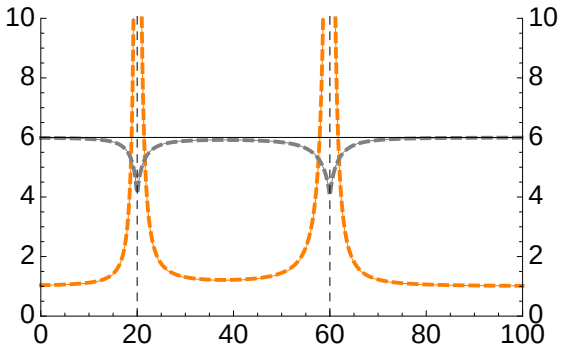
```
mya = .02;(* The selection coefficient in favour of A1. *)
myb = .4; (* The selection coefficient in favour of B1. *)
mym1 = 0.024; (* The backward migration rate from deme 1 to deme 2. *)
mym2 = mym1 / 100; (* The backward migration rate from deme 2 to deme 1. *)
myc1 = 0.01; (* The fraction of the population living in deme 1 (island). *)
myc2 = 1 - myc1; (* The fraction of the population living in deme 2 (continent). *)
myxA = 20;(*80;*) (* The posittion in map units of locus A. *)
myxB = 60;(*120;*) (* The posittion in map units of locus B. *)
mysf = 0.01; (* One map unit corresponds to r = mysf. *)
myRatio = 0.05; (* The minimum factor by which migration
 must be stronger than recombination for points to be plotted. *)
NTilde = 10^8; (* The size of the total population. *)
maxL = 100;(*200;*) (* The length of the chromosome in map units. *)
maxRate = 10;
colNe = GrayLevel[0.5];
plotScaling = 1;
logBase = 10;
MapThread[NTilde * #1 * #2 &, {{myc1, myc2}, {mym1, mym2}}]
plotCoalComb8 =
 Plot[{coalRateExactFunc[mya, myb, mym1, mym2, myc1, myc2, myxA, myxB, xC, mysf],
   coalRateFunc[mya, myb, mym1, mym2, myc1, myc2, myxA, myxB, xC, mysf, myRatio],
   plotScaling * Log[logBase, myc1 *
      NTilde / coalRateExactFunc[mya, myb, mym1, mym2, myc1, myc2, myxA, myxB, xC, mysf]],
   plotScaling * Log[logBase, myc1 * NTilde / coalRateFunc[mya, myb, mym1, mym2, myc1,
        myc2, myxA, myxB, xC, mysf, myRatio]], plotScaling * Log[logBase, myc1 * NTilde /
       coalRateFunc[0, 0, mym1, mym2, myc1, myc2, myxA, myxB, xC, mysf, myRatio]]},
  {xC, 0, maxL}, PlotRange → {{0, maxL}, {0, maxRate}}, Frame → True,
  FrameStyle → {{Black, Black}, {Black, Opacity[0]}},
  FrameTicks → {{All, All}, {All, All}},
  LabelStyle → {Directive[FontSize → 14], FontFamily → "Helvetica"}
  (*,FrameLabel→{{"Rate of coalescence","Log10 of coalescent Ne"},
    {"Position of neutral locus",""}}*),
  PlotStyle → {Orange, {Thick, Orange, Dashed}, colNe, {Thick, colNe, Dashed}, Black},
  Epilog → {{Black, Dashed, Line[{{myxA, 0}, {myxA, 1000}}]},
    {Black, Dashed, Line[{{myxB, 0}, {myxB, 1000}}]}}, ImageSize → 0.8 {360, 240}]
```

```
{24000., 23760.}
```



```
mya = .2;(* The selection coefficient in favour of A1. *)
myb = .4; (* The selection coefficient in favour of B1. *)
mym1 = 0.024; (* The backward migration rate from deme 1 to deme 2. *)
mym2 = mym1 / 100; (* The backward migration rate from deme 2 to deme 1. *)
myc1 = 0.01; (* The fraction of the population living in deme 1 (island). *)
myc2 = 1 - myc1; (* The fraction of the population living in deme 2 (continent). *)
myxA = 20;(*80;*) (* The posittion in map units of locus A. *)
myxB = 60;(*120;*) (* The posittion in map units of locus B. *)
mysf = 0.01; (* One map unit corresponds to r = mysf. *)
myRatio = 0.05; (* The minimum factor by which migration
 must be stronger than recombination for points to be plotted. *)
NTilde = 10^8; (* The size of the total population. *)
maxL = 100;(*200;*) (* The length of the chromosome in map units. *)
maxRate = 10;
colNe = GrayLevel[0.5];
plotScaling = 1;
logBase = 10;
MapThread[NTilde * #1 * #2 &, {{myc1, myc2}, {mym1, mym2}}]
plotCoalComb9 =
 Plot[{coalRateExactFunc[mya, myb, mym1, mym2, myc1, myc2, myxA, myxB, xC, mysf],
   coalRateFunc[mya, myb, mym1, mym2, myc1, myc2, myxA, myxB, xC, mysf, myRatio],
   plotScaling * Log[logBase, myc1 *
      NTilde / coalRateExactFunc[mya, myb, mym1, mym2, myc1, myc2, myxA, myxB, xC, mysf]],
   plotScaling * Log[logBase, myc1 * NTilde / coalRateFunc[mya, myb, mym1, mym2, myc1,
        myc2, myxA, myxB, xC, mysf, myRatio]], plotScaling * Log[logBase, myc1 * NTilde /
       coalRateFunc[0, 0, mym1, mym2, myc1, myc2, myxA, myxB, xC, mysf, myRatio]]},
  {xC, 0, maxL}, PlotRange → {{0, maxL}, {0, maxRate}}, Frame → True,
  FrameStyle → {{Black, Black}, {Black, Opacity[0]}},
  FrameTicks → {{All, All}, {All, All}},
  LabelStyle → {Directive[FontSize → 14], FontFamily → "Helvetica"}
  (*,FrameLabel→{{"Rate of coalescence","Log10 of coalescent Ne"},
    {"Position of neutral locus",""}}*),
  PlotStyle → {Orange, {Thick, Orange, Dashed}, colNe, {Thick, colNe, Dashed}, Black},
  Epilog → {{Black, Dashed, Line[{{myxA, 0}, {myxA, 1000}}]},
    {Black, Dashed, Line[{{myxB, 0}, {myxB, 1000}}]}}, ImageSize → 0.8 {360, 240}]
```

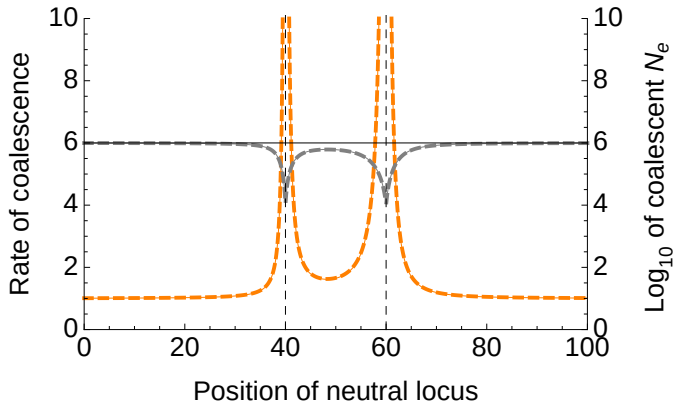
```
{24 000., 23 760.}
```



■ Various plots: $\mathcal{A}$ and $\mathcal{B}$ rather close, $c_1 = 0.01$, $m_1 / m_2 = 100$

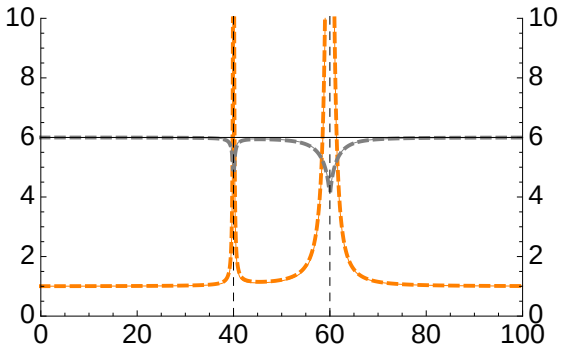
```
mya = .1; (* The selection coefficient in favour of A1. *)
myb = .4; (* The selection coefficient in favour of B1. *)
mym1 = 0.01; (* The backward migration rate from deme 1 to deme 2. *)
mym2 = mym1 / 100; (* The backward migration rate from deme 2 to deme 1. *)
myc1 = 0.01; (* The fraction of the population living in deme 1 (island). *)
myc2 = 1 - myc1; (* The fraction of the population living in deme 2 (continent). *)
myxA = 40; (*80;*) (* The posittion in map units of locus A. *)
myxB = 60; (*120;*) (* The posittion in map units of locus B. *)
mysf = 0.01; (* One map unit corresponds to r = mysf. *)
myRatio = 0.05; (* The minimum factor by which migration
 must be stronger than recombination for points to be plotted. *)
NTilde = 10^8; (* The size of the total population. *)
maxL = 100; (*200;*) (* The length of the chromosome in map units. *)
maxRate = 10;
colNe = GrayLevel[0.5];
plotScaling = 1;
logBase = 10;
MapThread[NTilde * #1 * #2 &, {{myc1, myc2}, {mym1, mym2}}]
plotCoalComb10 =
 Plot[{coalRateExactFunc[mya, myb, mym1, mym2, myc1, myc2, myxA, myxB, xC, mysf],
   coalRateFunc[mya, myb, mym1, mym2, myc1, myc2, myxA, myxB, xC, mysf, myRatio],
   plotScaling * Log[logBase, myc1 *
      NTilde / coalRateExactFunc[mya, myb, mym1, mym2, myc1, myc2, myxA, myxB, xC, mysf]],
   plotScaling * Log[logBase, myc1 * NTilde / coalRateFunc[mya, myb, mym1, mym2, myc1,
       myc2, myxA, myxB, xC, mysf, myRatio]], plotScaling * Log[logBase, myc1 * NTilde /
      coalRateFunc[0, 0, mym1, mym2, myc1, myc2, myxA, myxB, xC, mysf, myRatio]]},
  {xC, 0, maxL}, PlotRange → {{0, maxL}, {0, maxRate}}, Frame → True,
  FrameStyle → {{Black, Black}, {Black, Opacity[0]}},
  FrameTicks → {{All, All}, {All, All}},
  LabelStyle → {Directive[FontSize → 14], FontFamily → "Helvetica"},
  FrameLabel → {{"Rate of coalescence", "Log10 of coalescent Ne "},
    {"Position of neutral locus", ""}},
  PlotStyle → {Orange, {Thick, Orange, Dashed}, colNe, {Thick, colNe, Dashed}, Black},
  Epilog → {{Black, Dashed, Line[{{myxA, 0}, {myxA, 1000}}]},
    {Black, Dashed, Line[{{myxB, 0}, {myxB, 1000}}]}}, ImageSize → 1 {360, 240}]
```

```
{10 000., 9900.}
```



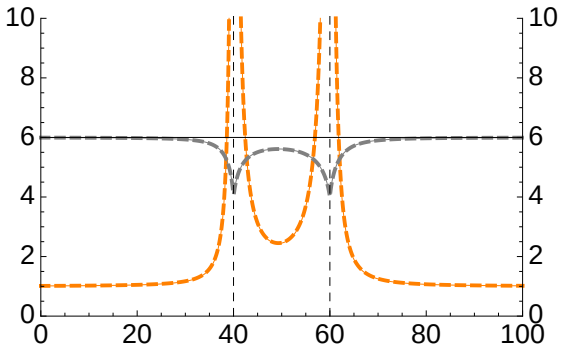
```
mya = .02;(* The selection coefficient in favour of A1. *)
myb = .4; (* The selection coefficient in favour of B1. *)
mym1 = 0.024; (* The backward migration rate from deme 1 to deme 2. *)
mym2 = mym1 / 100; (* The backward migration rate from deme 2 to deme 1. *)
myc1 = 0.01; (* The fraction of the population living in deme 1 (island). *)
myc2 = 1 - myc1; (* The fraction of the population living in deme 2 (continent). *)
myxA = 40;(*80;*) (* The posittion in map units of locus A. *)
myxB = 60;(*120;*) (* The posittion in map units of locus B. *)
mysf = 0.01; (* One map unit corresponds to r = mysf. *)
myRatio = 0.05; (* The minimum factor by which migration
 must be stronger than recombination for points to be plotted. *)
NTilde = 10^8; (* The size of the total population. *)
maxL = 100;(*200;*) (* The length of the chromosome in map units. *)
maxRate = 10;
colNe = GrayLevel[0.5];
plotScaling = 1;
logBase = 10;
MapThread[NTilde * #1 * #2 &, {{myc1, myc2}, {mym1, mym2}}]
plotCoalComb11 =
 Plot[{coalRateExactFunc[mya, myb, mym1, mym2, myc1, myc2, myxA, myxB, xC, mysf],
   coalRateFunc[mya, myb, mym1, mym2, myc1, myc2, myxA, myxB, xC, mysf, myRatio],
   plotScaling * Log[logBase, myc1 *
      NTilde / coalRateExactFunc[mya, myb, mym1, mym2, myc1, myc2, myxA, myxB, xC, mysf]],
   plotScaling * Log[logBase, myc1 * NTilde / coalRateFunc[mya, myb, mym1, mym2, myc1,
        myc2, myxA, myxB, xC, mysf, myRatio]], plotScaling * Log[logBase, myc1 * NTilde /
       coalRateFunc[0, 0, mym1, mym2, myc1, myc2, myxA, myxB, xC, mysf, myRatio]]},
  {xC, 0, maxL}, PlotRange → {{0, maxL}, {0, maxRate}}, Frame → True,
  FrameStyle → {{Black, Black}, {Black, Opacity[0]}},
  FrameTicks → {{All, All}, {All, All}},
  LabelStyle → {Directive[FontSize → 14], FontFamily → "Helvetica"}
  (*,FrameLabel→{{"Rate of coalescence","Log10 of coalescent Ne"},
    {"Position of neutral locus",""}}*),
  PlotStyle → {Orange, {Thick, Orange, Dashed}, colNe, {Thick, colNe, Dashed}, Black},
  Epilog → {{Black, Dashed, Line[{{myxA, 0}, {myxA, 1000}}]},
    {Black, Dashed, Line[{{myxB, 0}, {myxB, 1000}}]}}, ImageSize → 0.8 {360, 240}]
```

```
{24000., 23760.}
```



```
mya = .2; (* The selection coefficient in favour of A1. *)
myb = .4; (* The selection coefficient in favour of B1. *)
mym1 = 0.024; (* The backward migration rate from deme 1 to deme 2. *)
mym2 = mym1 / 100; (* The backward migration rate from deme 2 to deme 1. *)
myc1 = 0.01; (* The fraction of the population living in deme 1 (island). *)
myc2 = 1 - myc1; (* The fraction of the population living in deme 2 (continent). *)
myxA = 40;(*80;*) (* The posittion in map units of locus A. *)
myxB = 60;(*120;*) (* The posittion in map units of locus B. *)
mysf = 0.01; (* One map unit corresponds to r = mysf. *)
myRatio = 0.05; (* The minimum factor by which migration
 must be stronger than recombination for points to be plotted. *)
NTilde = 10^8; (* The size of the total population. *)
maxL = 100;(*200;*) (* The length of the chromosome in map units. *)
maxRate = 10;
colNe = GrayLevel[0.5];
plotScaling = 1;
logBase = 10;
MapThread[NTilde * #1 * #2 &, {{myc1, myc2}, {mym1, mym2}}]
plotCoalComb12 =
 Plot[{coalRateExactFunc[mya, myb, mym1, mym2, myc1, myc2, myxA, myxB, xC, mysf],
   coalRateFunc[mya, myb, mym1, mym2, myc1, myc2, myxA, myxB, xC, mysf, myRatio],
   plotScaling * Log[logBase, myc1 *
      NTilde / coalRateExactFunc[mya, myb, mym1, mym2, myc1, myc2, myxA, myxB, xC, mysf]],
   plotScaling * Log[logBase, myc1 * NTilde / coalRateFunc[mya, myb, mym1, mym2, myc1,
        myc2, myxA, myxB, xC, mysf, myRatio]], plotScaling * Log[logBase, myc1 * NTilde /
       coalRateFunc[0, 0, mym1, mym2, myc1, myc2, myxA, myxB, xC, mysf, myRatio]]},
  {xC, 0, maxL}, PlotRange → {{0, maxL}, {0, maxRate}}, Frame → True,
  FrameStyle → {{Black, Black}, {Black, Opacity[0]}},
  FrameTicks → {{All, All}, {All, All}},
  LabelStyle → {Directive[FontSize → 14], FontFamily → "Helvetica"}
  (*,FrameLabel→{{"Rate of coalescence","Log10 of coalescent Ne"},
    {"Position of neutral locus",""}}*),
  PlotStyle → {Orange, {Thick, Orange, Dashed}, colNe, {Thick, colNe, Dashed}, Black},
  Epilog → {{Black, Dashed, Line[{{myxA, 0}, {myxA, 1000}}]},
    {Black, Dashed, Line[{{myxB, 0}, {myxB, 1000}}]}}, ImageSize → 0.8 {360, 240}]
```

```
{24000., 23760.}
```



```
coalPlot2 =
 Labeled[GraphicsGrid[Partition[MapThread[Labeled[#1, #2, {{Top, Left}}, LabelStyle →
         {Directive[FontSize → 20, Bold], FontFamily → "Helvetica"},
       FrameMargins → {{12, 4}, {-2, -10}}] &, {{plotCoalComb8, plotCoalComb11,
       plotCoalComb9, plotCoalComb12}, CharacterRange["A", "D"]}], 2],
   ImageSize → 1.1 {600, 405}, AspectRatio → 0.7], {"Position of neutral locus",
   "Rate of coalescence", "Log10 of coalescent Ne"},
  {Bottom, Left, Right}, RotateLabel → True,
  LabelStyle → {Directive[FontSize → 20], FontFamily → "Helvetica"},
  FrameMargins → {{0, -2}, {5, 0}}]
```

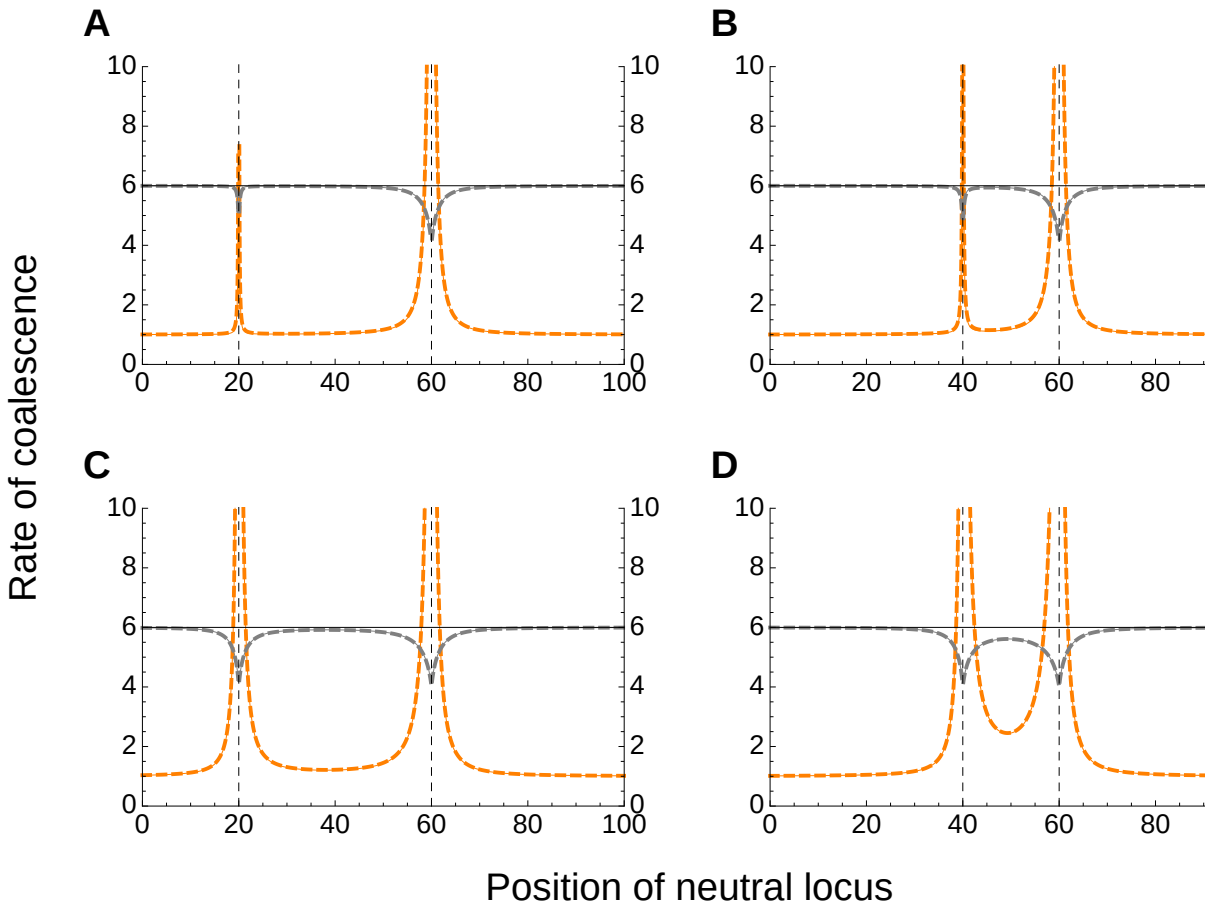


Figure 8: The effect on neutral coalescence of linkage to two sites at migration-selection balance. The rate of coalescence G (orange) and the coalescent effective size $\tilde{N}/G$ (gray) are given as a function of the position (in map units) of the neutral locus. Solid and dashed lines belong to values computed using the exact and approximate (Eq. 1 to 3) effective migration rate, respectively. One map unit (cM) is assumed to correspond to a recombination rate of $r = 0.01$ and the position of the sites under selection is indicated by vertical dashed lines. The total population size is $\tilde{N} = 10^8$, the fraction of the island is $c_1 = 0.001$ and the selection coefficient at locus $\mathcal{B}$ (position 60) is $b = 0.4$. (A) and (B) Migration rate to the island of the same order of magnitude as selection at locus $\mathcal{A}$: $a = 0.02$, $m_1 = 0.024$. (C) and (D) Weak immigration compared to selection at locus $\mathcal{A}$: $a = 0.2$, $m_1 = 0.024$. Throughout, $m_1/m_2 = 100$.

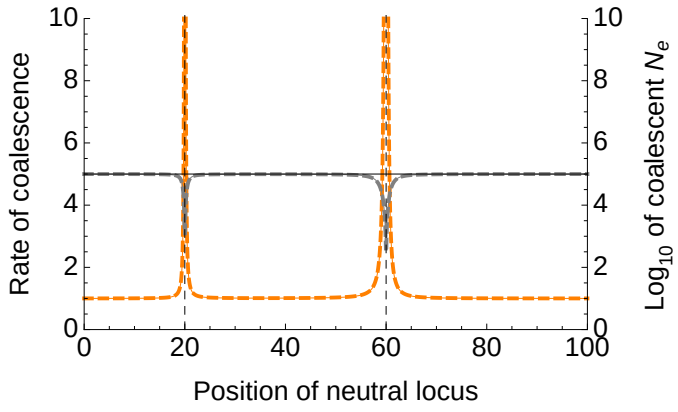
```
Export[PlotPath <> "coalRateAndNe_comb2.eps", coalPlot2, "EPS"]
```

```
/Users/Simon/Documents/LocAdD/results/130419/linkedNeutrSite/figures/
  coalRateAndNe_comb2.eps
```

■ Various plots: $\mathcal{A}$ and $\mathcal{B}$ far apart, $c_1 = 0.001$, $m_1 / m_2 = 1000$

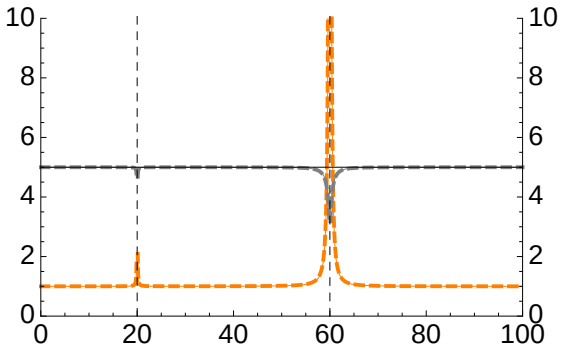
```
mya = .1; (* The selection coefficient in favour of A1. *)
myb = .4; (* The selection coefficient in favour of B1. *)
mym1 = 0.01; (* The backward migration rate from deme 1 to deme 2. *)
mym2 = mym1 / 1000; (* The backward migration rate from deme 2 to deme 1. *)
myc1 = 0.001; (* The fraction of the population living in deme 1 (island). *)
myc2 = 1 - myc1; (* The fraction of the population living in deme 2 (continent). *)
myxA = 20; (*80;*) (* The posittion in map units of locus A. *)
myxB = 60; (*120;*) (* The posittion in map units of locus B. *)
mysf = 0.01; (* One map unit corresponds to r = mysf. *)
myRatio = 0.05; (* The minimum factor by which migration
 must be stronger than recombination for points to be plotted. *)
NTilde = 10^8; (* The size of the total population. *)
maxL = 100; (*200;*) (* The length of the chromosome in map units. *)
maxRate = 10;
colNe = GrayLevel[0.5];
plotScaling = 1;
logBase = 10;
MapThread[NTilde * #1 * #2 &, {{myc1, myc2}, {mym1, mym2}}]
plotCoalComb13 =
 Plot[{coalRateExactFunc[mya, myb, mym1, mym2, myc1, myc2, myxA, myxB, xC, mysf],
   coalRateFunc[mya, myb, mym1, mym2, myc1, myc2, myxA, myxB, xC, mysf, myRatio],
   plotScaling * Log[logBase, myc1 *
      NTilde / coalRateExactFunc[mya, myb, mym1, mym2, myc1, myc2, myxA, myxB, xC, mysf]],
   plotScaling * Log[logBase, myc1 * NTilde / coalRateFunc[mya, myb, mym1, mym2, myc1,
       myc2, myxA, myxB, xC, mysf, myRatio]], plotScaling * Log[logBase, myc1 * NTilde /
      coalRateFunc[0, 0, mym1, mym2, myc1, myc2, myxA, myxB, xC, mysf, myRatio]]},
  {xC, 0, maxL}, PlotRange → {{0, maxL}, {0, maxRate}}, Frame → True,
  FrameStyle → {{Black, Black}, {Black, Opacity[0]}},
  FrameTicks → {{All, All}, {All, All}},
  LabelStyle → {Directive[FontSize → 14], FontFamily → "Helvetica"},
  FrameLabel → {{"Rate of coalescence", "Log10 of coalescent Ne"},
    {"Position of neutral locus", ""}},
  PlotStyle → {Orange, {Thick, Orange, Dashed}, colNe, {Thick, colNe, Dashed}, Black},
  Epilog → {{Black, Dashed, Line[{{myxA, 0}, {myxA, 1000}}]},
    {Black, Dashed, Line[{{myxB, 0}, {myxB, 1000}}]}}, ImageSize → 1 {360, 240}]
```

```
{1000., 999.}
```



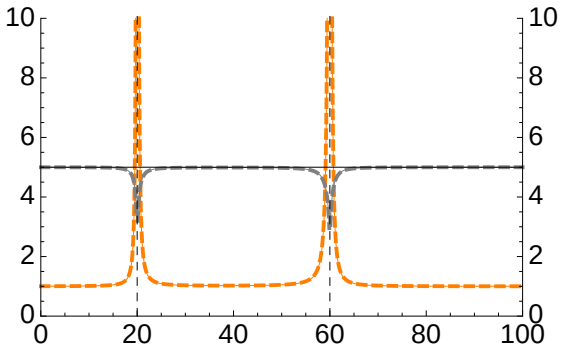
```
mya = .02;(* The selection coefficient in favour of A1. *)
myb = .4; (* The selection coefficient in favour of B1. *)
mym1 = 0.024; (* The backward migration rate from deme 1 to deme 2. *)
mym2 = mym1 / 1000; (* The backward migration rate from deme 2 to deme 1. *)
myc1 = 0.001; (* The fraction of the population living in deme 1 (island). *)
myc2 = 1 - myc1; (* The fraction of the population living in deme 2 (continent). *)
myxA = 20;(*80;*) (* The posittion in map units of locus A. *)
myxB = 60;(*120;*) (* The posittion in map units of locus B. *)
mysf = 0.01; (* One map unit corresponds to r = mysf. *)
myRatio = 0.05; (* The minimum factor by which migration
 must be stronger than recombination for points to be plotted. *)
NTilde = 10^8; (* The size of the total population. *)
maxL = 100;(*200;*) (* The length of the chromosome in map units. *)
maxRate = 10;
colNe = GrayLevel[0.5];
plotScaling = 1;
logBase = 10;
MapThread[NTilde * #1 * #2 &, {{myc1, myc2}, {mym1, mym2}}]
plotCoalComb14 =
 Plot[{coalRateExactFunc[mya, myb, mym1, mym2, myc1, myc2, myxA, myxB, xC, mysf],
   coalRateFunc[mya, myb, mym1, mym2, myc1, myc2, myxA, myxB, xC, mysf, myRatio],
   plotScaling * Log[logBase, myc1 *
      NTilde / coalRateExactFunc[mya, myb, mym1, mym2, myc1, myc2, myxA, myxB, xC, mysf]],
   plotScaling * Log[logBase, myc1 * NTilde / coalRateFunc[mya, myb, mym1, mym2, myc1,
        myc2, myxA, myxB, xC, mysf, myRatio]], plotScaling * Log[logBase, myc1 * NTilde /
       coalRateFunc[0, 0, mym1, mym2, myc1, myc2, myxA, myxB, xC, mysf, myRatio]]},
  {xC, 0, maxL}, PlotRange → {{0, maxL}, {0, maxRate}}, Frame → True,
  FrameStyle → {{Black, Black}, {Black, Opacity[0]}},
  FrameTicks → {{All, All}, {All, All}},
  LabelStyle → {Directive[FontSize → 14], FontFamily → "Helvetica"}
  (*,FrameLabel→{{"Rate of coalescence","Log10 of coalescent Ne"},
    {"Position of neutral locus",""}}*),
  PlotStyle → {Orange, {Thick, Orange, Dashed}, colNe, {Thick, colNe, Dashed}, Black},
  Epilog → {{Black, Dashed, Line[{{myxA, 0}, {myxA, 1000}}]},
    {Black, Dashed, Line[{{myxB, 0}, {myxB, 1000}}]}}, ImageSize → 0.8 {360, 240}]
```

```
{2400., 2397.6}
```



```
mya = .2;(* The selection coefficient in favour of A1. *)
myb = .4; (* The selection coefficient in favour of B1. *)
mym1 = 0.024; (* The backward migration rate from deme 1 to deme 2. *)
mym2 = mym1 / 1000; (* The backward migration rate from deme 2 to deme 1. *)
myc1 = 0.001; (* The fraction of the population living in deme 1 (island). *)
myc2 = 1 - myc1; (* The fraction of the population living in deme 2 (continent). *)
myxA = 20;(*80;*) (* The posittion in map units of locus A. *)
myxB = 60;(*120;*) (* The posittion in map units of locus B. *)
mysf = 0.01; (* One map unit corresponds to r = mysf. *)
myRatio = 0.05; (* The minimum factor by which migration
 must be stronger than recombination for points to be plotted. *)
NTilde = 10^8; (* The size of the total population. *)
maxL = 100;(*200;*) (* The length of the chromosome in map units. *)
maxRate = 10;
colNe = GrayLevel[0.5];
plotScaling = 1;
logBase = 10;
MapThread[NTilde * #1 * #2 &, {{myc1, myc2}, {mym1, mym2}}]
plotCoalComb15 =
 Plot[{coalRateExactFunc[mya, myb, mym1, mym2, myc1, myc2, myxA, myxB, xC, mysf],
   coalRateFunc[mya, myb, mym1, mym2, myc1, myc2, myxA, myxB, xC, mysf, myRatio],
   plotScaling * Log[logBase, myc1 *
      NTilde / coalRateExactFunc[mya, myb, mym1, mym2, myc1, myc2, myxA, myxB, xC, mysf]],
   plotScaling * Log[logBase, myc1 * NTilde / coalRateFunc[mya, myb, mym1, mym2, myc1,
        myc2, myxA, myxB, xC, mysf, myRatio]], plotScaling * Log[logBase, myc1 * NTilde /
       coalRateFunc[0, 0, mym1, mym2, myc1, myc2, myxA, myxB, xC, mysf, myRatio]]},
  {xC, 0, maxL}, PlotRange → {{0, maxL}, {0, maxRate}}, Frame → True,
  FrameStyle → {{Black, Black}, {Black, Opacity[0]}},
  FrameTicks → {{All, All}, {All, All}},
  LabelStyle → {Directive[FontSize → 14], FontFamily → "Helvetica"}
  (*,FrameLabel→{{"Rate of coalescence","Log10 of coalescent Ne"},
    {"Position of neutral locus",""}}*),
  PlotStyle → {Orange, {Thick, Orange, Dashed}, colNe, {Thick, colNe, Dashed}, Black},
  Epilog → {{Black, Dashed, Line[{{myxA, 0}, {myxA, 1000}}]},
    {Black, Dashed, Line[{{myxB, 0}, {myxB, 1000}}]}}, ImageSize → 0.8 {360, 240}]
```

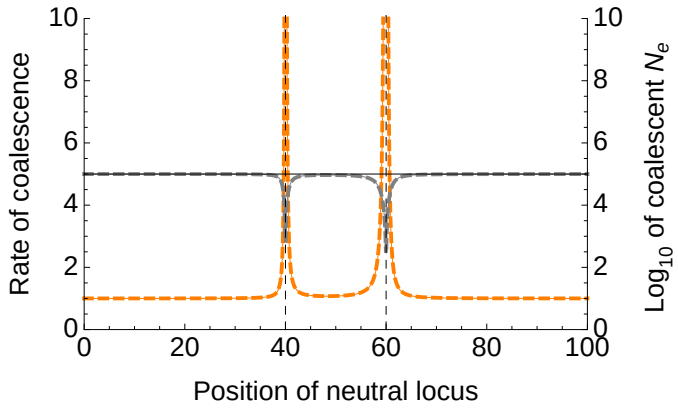
```
{2400., 2397.6}
```



- Various plots: $\mathcal{A}$ and $\mathcal{B}$ rather close, $c_1 = 0.001$, $m_1 / m_2 = 1000$

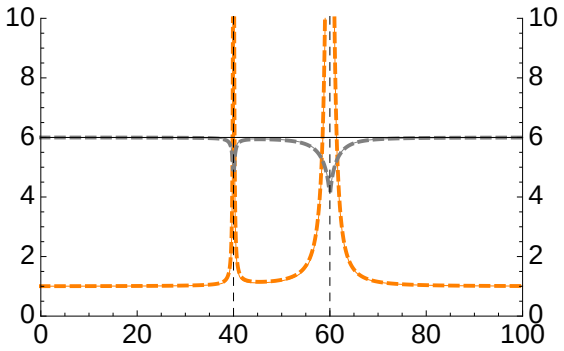
```
mya = .1; (* The selection coefficient in favour of A1. *)
myb = .4; (* The selection coefficient in favour of B1. *)
mym1 = 0.01; (* The backward migration rate from deme 1 to deme 2. *)
mym2 = mym1 / 1000; (* The backward migration rate from deme 2 to deme 1. *)
myc1 = 0.001; (* The fraction of the population living in deme 1 (island). *)
myc2 = 1 - myc1; (* The fraction of the population living in deme 2 (continent). *)
myxA = 40; (*80;*) (* The posittion in map units of locus A. *)
myxB = 60; (*120;*) (* The posittion in map units of locus B. *)
mysf = 0.01; (* One map unit corresponds to r = mysf. *)
myRatio = 0.05; (* The minimum factor by which migration
 must be stronger than recombination for points to be plotted. *)
NTilde = 10^8; (* The size of the total population. *)
maxL = 100; (*200;*) (* The length of the chromosome in map units. *)
maxRate = 10;
colNe = GrayLevel[0.5];
plotScaling = 1;
logBase = 10;
MapThread[NTilde * #1 * #2 &, {{myc1, myc2}, {mym1, mym2}}]
plotCoalComb16 =
 Plot[{coalRateExactFunc[mya, myb, mym1, mym2, myc1, myc2, myxA, myxB, xC, mysf],
   coalRateFunc[mya, myb, mym1, mym2, myc1, myc2, myxA, myxB, xC, mysf, myRatio],
   plotScaling * Log[logBase, myc1 *
      NTilde / coalRateExactFunc[mya, myb, mym1, mym2, myc1, myc2, myxA, myxB, xC, mysf]],
   plotScaling * Log[logBase, myc1 * NTilde / coalRateFunc[mya, myb, mym1, mym2, myc1,
       myc2, myxA, myxB, xC, mysf, myRatio]], plotScaling * Log[logBase, myc1 * NTilde /
      coalRateFunc[0, 0, mym1, mym2, myc1, myc2, myxA, myxB, xC, mysf, myRatio]]},
  {xC, 0, maxL}, PlotRange → {{0, maxL}, {0, maxRate}}, Frame → True,
  FrameStyle → {{Black, Black}, {Black, Opacity[0]}},
  FrameTicks → {{All, All}, {All, All}},
  LabelStyle → {Directive[FontSize → 14], FontFamily → "Helvetica"},
  FrameLabel → {{"Rate of coalescence", "Log10 of coalescent Ne "},
    {"Position of neutral locus", ""}},
  PlotStyle → {Orange, {Thick, Orange, Dashed}, colNe, {Thick, colNe, Dashed}, Black},
  Epilog → {{Black, Dashed, Line[{{myxA, 0}, {myxA, 1000}}]},
    {Black, Dashed, Line[{{myxB, 0}, {myxB, 1000}}]}}, ImageSize → 1 {360, 240}]
```

```
{1000., 999.}
```



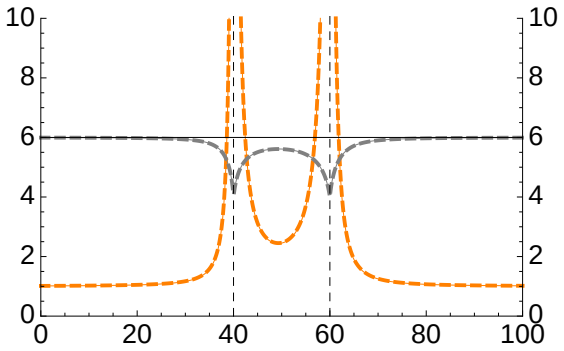
```
mya = .02;(* The selection coefficient in favour of A1. *)
myb = .4; (* The selection coefficient in favour of B1. *)
mym1 = 0.024; (* The backward migration rate from deme 1 to deme 2. *)
mym2 = mym1 / 100; (* The backward migration rate from deme 2 to deme 1. *)
myc1 = 0.01; (* The fraction of the population living in deme 1 (island). *)
myc2 = 1 - myc1; (* The fraction of the population living in deme 2 (continent). *)
myxA = 40;(*80;*) (* The posittion in map units of locus A. *)
myxB = 60;(*120;*) (* The posittion in map units of locus B. *)
mysf = 0.01; (* One map unit corresponds to r = mysf. *)
myRatio = 0.05; (* The minimum factor by which migration
 must be stronger than recombination for points to be plotted. *)
NTilde = 10^8; (* The size of the total population. *)
maxL = 100;(*200;*) (* The length of the chromosome in map units. *)
maxRate = 10;
colNe = GrayLevel[0.5];
plotScaling = 1;
logBase = 10;
MapThread[NTilde * #1 * #2 &, {{myc1, myc2}, {mym1, mym2}}]
plotCoalComb17 =
 Plot[{coalRateExactFunc[mya, myb, mym1, mym2, myc1, myc2, myxA, myxB, xC, mysf],
   coalRateFunc[mya, myb, mym1, mym2, myc1, myc2, myxA, myxB, xC, mysf, myRatio],
   plotScaling * Log[logBase, myc1 *
      NTilde / coalRateExactFunc[mya, myb, mym1, mym2, myc1, myc2, myxA, myxB, xC, mysf]],
   plotScaling * Log[logBase, myc1 * NTilde / coalRateFunc[mya, myb, mym1, mym2, myc1,
        myc2, myxA, myxB, xC, mysf, myRatio]], plotScaling * Log[logBase, myc1 * NTilde /
       coalRateFunc[0, 0, mym1, mym2, myc1, myc2, myxA, myxB, xC, mysf, myRatio]]},
  {xC, 0, maxL}, PlotRange → {{0, maxL}, {0, maxRate}}, Frame → True,
  FrameStyle → {{Black, Black}, {Black, Opacity[0]}},
  FrameTicks → {{All, All}, {All, All}},
  LabelStyle → {Directive[FontSize → 14], FontFamily → "Helvetica"}
  (*,FrameLabel→{{"Rate of coalescence","Log10 of coalescent Ne"},
    {"Position of neutral locus",""}}*),
  PlotStyle → {Orange, {Thick, Orange, Dashed}, colNe, {Thick, colNe, Dashed}, Black},
  Epilog → {{Black, Dashed, Line[{{myxA, 0}, {myxA, 1000}}]},
    {Black, Dashed, Line[{{myxB, 0}, {myxB, 1000}}]}}, ImageSize → 0.8 {360, 240}]
```

```
{24000., 23760.}
```



```
mya = .2;(* The selection coefficient in favour of A1. *)
myb = .4; (* The selection coefficient in favour of B1. *)
mym1 = 0.024; (* The backward migration rate from deme 1 to deme 2. *)
mym2 = mym1 / 100; (* The backward migration rate from deme 2 to deme 1. *)
myc1 = 0.01; (* The fraction of the population living in deme 1 (island). *)
myc2 = 1 - myc1; (* The fraction of the population living in deme 2 (continent). *)
myxA = 40;(*80;*) (* The posittion in map units of locus A. *)
myxB = 60;(*120;*) (* The posittion in map units of locus B. *)
mysf = 0.01; (* One map unit corresponds to r = mysf. *)
myRatio = 0.05; (* The minimum factor by which migration
 must be stronger than recombination for points to be plotted. *)
NTilde = 10^8; (* The size of the total population. *)
maxL = 100;(*200;*) (* The length of the chromosome in map units. *)
maxRate = 10;
colNe = GrayLevel[0.5];
plotScaling = 1;
logBase = 10;
MapThread[NTilde * #1 * #2 &, {{myc1, myc2}, {mym1, mym2}}]
plotCoalComb18 =
 Plot[{coalRateExactFunc[mya, myb, mym1, mym2, myc1, myc2, myxA, myxB, xC, mysf],
   coalRateFunc[mya, myb, mym1, mym2, myc1, myc2, myxA, myxB, xC, mysf, myRatio],
   plotScaling * Log[logBase, myc1 *
      NTilde / coalRateExactFunc[mya, myb, mym1, mym2, myc1, myc2, myxA, myxB, xC, mysf]],
   plotScaling * Log[logBase, myc1 * NTilde / coalRateFunc[mya, myb, mym1, mym2, myc1,
        myc2, myxA, myxB, xC, mysf, myRatio]], plotScaling * Log[logBase, myc1 * NTilde /
       coalRateFunc[0, 0, mym1, mym2, myc1, myc2, myxA, myxB, xC, mysf, myRatio]]},
  {xC, 0, maxL}, PlotRange → {{0, maxL}, {0, maxRate}}, Frame → True,
  FrameStyle → {{Black, Black}, {Black, Opacity[0]}},
  FrameTicks → {{All, All}, {All, All}},
  LabelStyle → {Directive[FontSize → 14], FontFamily → "Helvetica"}
  (*,FrameLabel→{{"Rate of coalescence","Log10 of coalescent Ne"},
    {"Position of neutral locus",""}}*),
  PlotStyle → {Orange, {Thick, Orange, Dashed}, colNe, {Thick, colNe, Dashed}, Black},
  Epilog → {{Black, Dashed, Line[{{myxA, 0}, {myxA, 1000}}]},
    {Black, Dashed, Line[{{myxB, 0}, {myxB, 1000}}]}}, ImageSize → 0.8 {360, 240}]
```

```
{24 000., 23 760.}
```



```
coalPlot2 =
 Labeled[GraphicsGrid[Partition[MapThread[Labeled[#1, #2, {{Top, Left}}, LabelStyle →
         {Directive[FontSize → 20, Bold], FontFamily → "Helvetica"},
       FrameMargins → {{12, 4}, {-2, -10}}] &, {{plotCoalComb14, plotCoalComb17,
       plotCoalComb15, plotCoalComb18}, CharacterRange["A", "D"]}], 2],
   ImageSize → 1.1 {600, 405}, AspectRatio → 0.7], {"Position of neutral locus",
   "Rate of coalescence", "Log10 of coalescent Ne"},
  {Bottom, Left, Right}, RotateLabel → True,
  LabelStyle → {Directive[FontSize → 20], FontFamily → "Helvetica"},
  FrameMargins → {{0, -2}, {5, 0}}]
```

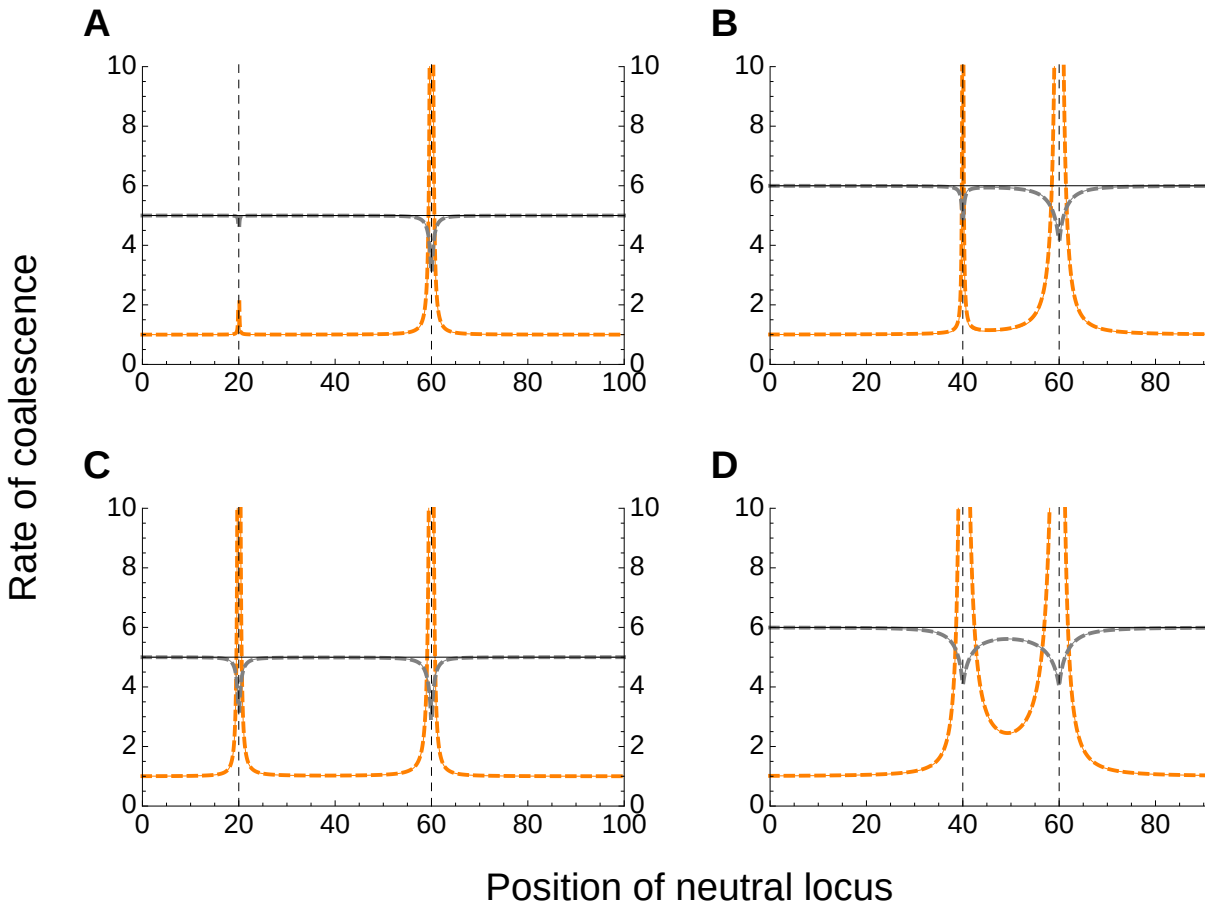


Figure 8: The effect on neutral coalescence of linkage to two sites at migration-selection balance. The rate of coalescence G (orange) and the coalescent effective size $\tilde{N}/G$ (gray) are given as a function of the position (in map units) of the neutral locus. Solid and dashed lines belong to values computed using the exact and approximate (Eq. 1 to 3) effective migration rate, respectively. One map unit (cM) is assumed to correspond to a recombination rate of $r = 0.01$ and the position of the sites under selection is indicated by vertical dashed lines. The total population size is $\tilde{N} = 10^8$, the fraction of the island is $c_1 = 0.001$ and the selection coefficient at locus $\mathcal{B}$ (position 60) is $b = 0.4$. (A) and (B) Migration rate to the island of the same order of magnitude as selection at locus $\mathcal{A}$: $a = 0.02$, $m_1 = 0.024$. (C) and (D) Weak immigration compared to selection at locus $\mathcal{A}$: $a = 0.2$, $m_1 = 0.024$. Throughout, $m_1/m_2 = 1000$.

```
Export[PlotPath <> "coalRateAndNe_comb3.eps", coalPlot3, "EPS"]
```

```
/Users/Simon/Documents/LocAdD/results/130419/linkedNeutrSite/figures/
  coalRateAndNe_comb3.eps
```

Functions and further initialisation cells

- **Sojourn-time density and absorption time**

In[261]:=
```
ψOLMNMonomFunc[μ_, n_] := (1 - n)^(-2 μ)
```

- **Stationary distribution of allele frequencies**
- **Coalescence**