

Evolutionary dynamics of neutral phenotypes under DNA substitution models

Deriving the central quantities used in the Appendix with Mathematica

Shadi Zabad and Alan M Moses

```
In[234]:= SetOptions[SelectedNotebook[], PrintingStyleEnvironment -> "Printout", ShowSyntaxStyles -> True]
```

Single Locus Statistics

Section 1: Defining the central quantities of our model

We start by defining the central quantities that constitute our model. This includes the single-locus transition probabilities:

```
In[196]:= (* The probability of transition to the k-th allele in a time interval t *)
P01[t_, pi_] := pi*(1 - Exp[-u*t])
(* The probability that the k-th allele remains fixed in a time interval t *)
P11[t_, pi_] := Exp[-u*t] + pi*(1 - Exp[-u*t])
```

As well as the probability of the k-th allele being fixed at any point in time 't':

```
In[198]:= P[t_, p0_, pi_] := p0*Exp[-u*t] + pi*(1 - Exp[-u*t])
```

For the next two sections, we need to define the probabilities for two alleles, indexed by 'k' and 'm'. To simplify notation, we define the following expressions:

```
In[199]:= (* The equilibrium probability of the k-th allele *)
pik = Subscript[pi, k];
(* The equilibrium probability of the m-th allele *)
pim = Subscript[pi, m];
(* The initial probability of the k-th allele *)
p0k = Subscript[p0, k];
(* The initial probability of the m-th allele *)
p0m = Subscript[p0, m];

(* The probability of observing the k-th allele at any point in time *)
Ptk = P[t, p0k, pik];
(* The probability of observing the m-th allele at any point in time *)
Ptm = P[t, p0m, pim];
```

Section 2: Deriving the covariance in allelic state between two descendant lineages (same allele)

In Section 3.1 of the Appendix, we defined the covariance in the allelic state in two descendant lineages that diverged at time 't'

and then each evolved independently for `s` and `r` time units, respectively. To arrive at an expression for this covariance, first we write out the expression for the joint probability:

```
In[205]:= jointProbSameAllele = Ptk*P11[s, pik]*P11[r, pik] + (1 - Ptk)*P01[s, pik]*P01[r, pik];
FullSimplify[jointProbSameAllele]
```

```
Out[206]= e-(r+s+t) u (p0k (1 + (-2 + er u + es u) πk) - πk (1 - et u + (-2 + er u + es u + et u - e(r+s+t) u) πk))
```

And then write the covariance using the standard definition and simplify:

```
In[207]:= covarianceSameAllele = jointProbSameAllele - P[t + s, p0k, pik]*P[t + r, p0k, pik];
FullSimplify[covarianceSameAllele]
```

```
Out[208]= -e-(r+s+2 t) u (p0k + et u (-1 + πk) - πk) (p0k + (-1 + et u) πk)
```

The above expression resembles some of the central quantities that we defined in Section 1. In fact, if we re-arrange by distributing the $\text{Exp}[-2ut]$ term inside the two parentheses, we see that the second parenthesis corresponds to the probability of observing the k -th allele at time t and the first corresponds to the complement of that probability. This is an intuitive and appealing result, since it says that the covariance is equivalent to the Bernoulli-like variance at time t multiplied by an exponential decay term that is proportional to the time elapsed between the two descendant lineages.

Section 3: Deriving the covariance in allelic state between two descendant lineages (two different alleles)

In section 3.1 of the Appendix, we also defined the covariance between two different alleles, indexed by k and m , in two descendant lineages that diverged at time t and then each evolved independently for s and r time units, respectively. To arrive at an expression for this covariance, first we write out and simplify the expression for the joint probability:

```
In[209]:= jointProbTwoAlleles = Ptk*P11[s, pik]*P01[r, pim] + Ptm*P01[s, pik]*P11[r, pim] +
(1 - Ptk - Ptm)*P01[s, pik]*P01[r, pim];
FullSimplify[jointProbTwoAlleles]
```

And then write the covariance using the standard definition and simplify:

```
In[209]:= covarianceTwoAlleles = jointProbTwoAlleles - P[t + s, p0k, pik]*P[t + r, p0m, pim];
FullSimplify[covarianceTwoAlleles]
```

```
Out[210]= -e-(r+s+2 t) u (p0k + (-1 + et u) πk) (p0m + (-1 + et u) πm)
```

Similar to our remarks above, this expression can also be expressed in terms of the central quantities defined above. In this case, once we distribute the $\text{Exp}[-2ut]$ terms, we obtain the probability of observing the k -th allele at time t in the first parenthesis and the probability of observing the m -th allele in the second parenthesis. This result is also intuitive, since it conveys the notion that the covariance is equivalent to the Multinomial-like variance at time t multiplied by an exponential decay term that is proportional to the time elapsed between the two descendant lineages.

Mean Phenotype Statistics

Section 4: Demonstration of the idea of “lumping” alleles that share effect sizes

In this section, our aim is to show that when the effect sizes are shared across different alleles, then it is possible to simply “lump” or group those alleles into a single allele. We will show this heuristically with examples rather than formally with proofs.

First, we start by defining the central quantities that we need for any number of alleles `K`:

```
In[211]:= (* Number of alleles *)
K = 6;
(* Unique effect sizes *)
uniqueBeta = {Subscript[b, 1], Subscript[b, 2]};
(* Number of unique effect sizes *)
uniqueK = Length[uniqueBeta];
(* Randomly assign the unique effect sizes to the K alleles *)
betas = RandomChoice[uniqueBeta, K];
(* Generate expressions for the equilibrium probabilities *)
pis = Table[Subscript[π, k], {k, Range[K]}];
(* Generate expressions for the initial probabilities *)
p0s = Table[Subscript[p0, k], {k, Range[K]}];

(* The betas from this random draw are: *)
betas
```

```
Out[217]= {b2, b2, b1, b1, b2, b2}
```

Given this, we can write the probability of observing each of the K alleles at any point in time `t` as:

```
In[218]:= Pt = Table[P[t, p0s[[k]], pis[[k]]], {k, Range[K]}];
```

Next, to illustrate the idea of “lumping” alleles, we are going to reconstruct the expressions above by summing them for each unique effect size. Thus instead of having `K` alleles, we will have a number of alleles that is equivalent to the cardinality of `uniqueBeta`. In this particular example, we will group a total of 6 alleles into two groups having effect sizes `b1` and `b2`:

```
In[219]:= (* Indices of each effect size in `betas` *)
indices = Table[Flatten[Position[betas, k]], {k, uniqueBeta}];
(* Grouping equilibrium probabilities *)
groupedPis = Table[Total[Table[pis[[k]], {k, g}]], {g, indices}];
(* Grouping initial probabilities *)
groupedP0s = Table[Total[Table[p0s[[k]], {k, g}]], {g, indices}];
(* Probabilities of observing the lumped alleles at any point in time *)
groupedPt = Table[P[t, groupedP0s[[k]], groupedPis[[k]]], {k, Range[uniqueK]}];

(* This results in lumped alleles with the following probabilities: *)
{groupedPis, groupedP0s}
```

```
Out[223]= {{π3 + π4, π1 + π2 + π5 + π6}, {p03 + p04, p01 + p02 + p05 + p06}}
```

Finally, to assist with deriving the covariance dynamics, we construct functions that define the covariances in allelic state:

```
In[224]:= (* Covariance of same allele in two descendant lineages *)
covarianceSameAlleleF[p0_, pi_] := P[t, p0, pi]*P11[s, pi]*P11[r, pi] +
    (1 - P[t, p0, pi])*P01[s, pi]*P01[r, pi] -
    P[t + s, p0, pi]*P[t + r, p0, pi];
(* Covariance of two different alleles in two descendant lineages *)
covarianceTwoAllelesF[p0k_, pik_, p0m_, pim_] := P[t, p0k, pik]*P11[s, pik]*P01[r, pim] +
    P[t, p0m, pim]*P01[s, pik]*P11[r, pim] +
    (1 - P[t, p0k, pik] - P[t, p0m, pim])*P01[s, pik]*P01[r, pim] -
    P[t + s, p0k, pik]*P[t + r, p0m, pim];
```

Test that the functions above return sensible results:

```
In[226]:= FullSimplify[covarianceSameAlleleF[p0k, pik]]
```

```
Out[226]= -e^{-(r+s+2t)u} (p0k + e^{tu} (-1 + \pi_k) - \pi_k) (p0k + (-1 + e^{tu}) \pi_k)
```

```
In[227]:= FullSimplify[covarianceTwoAllelesF[p0k, pik, p0m, pim]]
```

```
Out[227]= -e^{-(r+s+2t)u} (p0k + (-1 + e^{tu}) \pi_k) (p0m + (-1 + e^{tu}) \pi_m)
```

The Ensemble Mean Dynamics

Here, our aim is to show that the mean dynamics under the full model as well as the simpler “lumped” model are identical. Note that we only consider a single locus, since we assume that loci are independent and are subject to the same underlying dynamics. First, we evaluate the full and lumped models according to Equation 14 in the appendix:

```
In[228]:= (* The ensemble mean under the full model *)
meanFullModel = Total[betas*Pt];
(* The ensemble mean under the lumped model *)
meanGroupedModel = Total[uniqueBeta*groupedPt];
```

And then we check that they are algebraically equivalent with Mathematica:

```
In[230]:= FullSimplify[meanFullModel == meanGroupedModel]
```

```
Out[230]= True
```

The Ensemble Covariance Dynamics

In this section, our aim is to show that the covariance dynamics under the full model as well as the simpler “lumped” model are identical. Note again we only consider a single locus, since the covariance between different loci is assumed to be zero by default and that different loci are subject to the same underlying dynamics. First we evaluate the covariance under the full and lumped models according to Equation 15 in the appendix:

In[231]:=

```

(* The ensemble covariance under the full model *)
covarianceFullModel = Total[
  (* Outer loop *)
  Table[
    betas[[k]]^2 * covarianceSameAlleleF[p0s[[k]], pis[[k]]] +
    Total[
      (* Inner loop *)
      Table[
        betas[[k]]*betas[[m]]*
        covarianceTwoAllelesF[p0s[[k]], pis[[k]], p0s[[m]], pis[[m]]],
        {m, Select[Range[K], # ≠ k &]}
      ]
    ],
    {k, Range[K]}
  ]
];

(* The ensemble covariance under the lumped model *)
covarianceGroupedModel = Total[
  (* Outer loop *)
  Table[
    uniqueBeta[[k]]^2 * covarianceSameAlleleF[groupedP0s[[k]], groupedPis[[k]]] +
    Total[
      (* Inner loop *)
      Table[
        uniqueBeta[[k]]*uniqueBeta[[m]]*
        covarianceTwoAllelesF[groupedP0s[[k]], groupedPis[[k]],
          groupedP0s[[m]], groupedPis[[m]]],
        {m, Select[Range[uniqueK], # ≠ k &]}
      ]
    ],
    {k, Range[uniqueK]}
  ]
];

```

And then we check that they are algebraically equivalent with Mathematica (this may take a while to run, depending on the number of alleles):

In[231]:=

```
FullSimplify[covarianceFullModel == covarianceGroupedModel]
```

Out[231]= True