Optimization of gene expression by natural selection
Trevor Bedford and Daniel L. Hartl

Maximum-likelihood estimation of OU parameters

Here we use the Ornstein-Uhlenbeck process to model gene expression divergence. This document gives a brief tutorial in gene-specific parameter estimation and hypothesis testing.

First, refer to Maximum-likelihood estimation of OU parameters section of Methods.

R code is provided, but the methods will work with any statistical / programming software package. The package mvtnorm (http://cran.r-project.org/web/packages/mvtnorm/index.html) is used to compute the PDF of multivariate normal distributions. Once installed this package is loaded with the command:

```
library(mvtnorm);
```

Expression values

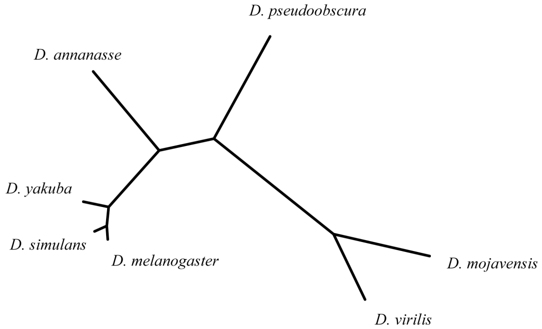
Using as an example gene FBgn0011768 with expression vector

```
expr <- c(1.288733,0.7893953,0.814975,0.6608836,0.3954609,0.7329379,0.836787)
```

in *dmel*, *dsim*, *dyak*, *dana*, *dpse*, *dmoj*, and *dvir*.

Distance (*t*) matrices

The first step is to estimate the evolutionary time points at which gene expression values are sampled. Here, we assume this corresponds to the *Drosophila* phylogeny, with branch-lengths proportional to amino acid distance.



The tree shown here is: (((((dmel: 0.008306, dsim: 0.008237): 0.011678, dyak: 0.015866): 0.046511, dana: 0.063118): 0.034376, dpse: 0.071528): 0.093932, dmoj: 0.060454, dvir: 0.044506).

We take the starting point for evolution along this tree as *D. melanogaster.* The distance from *D. melangaster* to each pair of the other six species is:

```
tdmatrix <- matrix(c(0.016543, 0.044087, 0.13785, 0.180636, 0.263494, 0.247546,
0.044087, 0.03585, 0.145479, 0.188265, 0.271123, 0.255175, 0.13785, 0.145479,
0.129613, 0.235517, 0.318375, 0.302427, 0.180636, 0.188265, 0.235517, 0.172399,
0.326785, 0.310837, 0.263494, 0.271123, 0.318375, 0.326785, 0.255257, 0.299763,
0.247546, 0.255175, 0.302427, 0.310837, 0.299763, 0.239309),nrow=6,byrow=TRUE);
```

```
> tdmatrix
         [,1]     [,2]     [,3]     [,4]     [,5]     [,6]
[1,] 0.016543 0.044087 0.137850 0.180636 0.263494 0.247546
[2,] 0.044087 0.035850 0.145479 0.188265 0.271123 0.255175
[3,] 0.137850 0.145479 0.129613 0.235517 0.318375 0.302427
[4,] 0.180636 0.188265 0.235517 0.172399 0.326785 0.310837
[5,] 0.263494 0.271123 0.318375 0.326785 0.255257 0.299763
[6,] 0.247546 0.255175 0.302427 0.310837 0.299763 0.239309
```

So, the total distance from *dmel* to *dsim* is 0.017, while the total distance in the *dmel*, *dsim*, *dyak* tree is 0.044.

The diagonal of the `tdmatrix` gives t_i.

```
meanslist <- diag(tdmatrix);
```

```
> meanslist
[1] 0.016543 0.035850 0.129613 0.172399 0.255257 0.239309
```

The shared distance from *dmel* to each pair of species is:

```
sdmatrix <- matrix(c(0.016543, 0.008306, 0.008306, 0.008306, 0.008306, 0.008306,
0.008306, 0.03585, 0.019984, 0.019984, 0.019984, 0.019984, 0.008306,
0.019984, 0.129613, 0.066495, 0.066495, 0.066495, 0.008306, 0.019984,
0.066495, 0.172399, 0.100871, 0.100871, 0.008306, 0.019984, 0.066495,
0.100871, 0.255257, 0.194803, 0.008306, 0.019984, 0.066495, 0.100871,
0.194803, 0.239309),nrow=6,byrow=TRUE);
```

```
> sdmatrix
         [,1]     [,2]     [,3]     [,4]     [,5]     [,6]
[1,] 0.016543 0.008306 0.008306 0.008306 0.008306 0.008306
[2,] 0.008306 0.035850 0.019984 0.019984 0.019984 0.019984
[3,] 0.008306 0.019984 0.129613 0.066495 0.066495 0.066495
[4,] 0.008306 0.019984 0.066495 0.172399 0.100871 0.100871
[5,] 0.008306 0.019984 0.066495 0.100871 0.255257 0.194803
[6,] 0.008306 0.019984 0.066495 0.100871 0.194803 0.239309
```

Dsim and *dyak* share evolution along the *dmel* branch of the phylogeny giving their shared distance as 0.008.

Tracing the total evolutionary distance going from *dsim* through *dmel* to *dyak* requires counting the *dsim* and *dyak* specific branches once and the *dmel* specific branch twice. This can be accomplished by:

```
> cmatrix <- sdmatrix+tdmatrix;

> cmatrix
         [,1]     [,2]     [,3]     [,4]     [,5]     [,6]
[1,] 0.033086 0.052393 0.146156 0.188942 0.271800 0.255852
[2,] 0.052393 0.071700 0.165463 0.208249 0.291107 0.275159
[3,] 0.146156 0.165463 0.259226 0.302012 0.384870 0.368922
[4,] 0.188942 0.208249 0.302012 0.344798 0.427656 0.411708
[5,] 0.271800 0.291107 0.384870 0.427656 0.510514 0.494566
[6,] 0.255852 0.275159 0.368922 0.411708 0.494566 0.478618
```

The `cmatrix` gives $t_i + t_j$, while the `sdmatrix` gives $s_{i/j}$.

Multivariate normal PDF

The mean value after time t depends upon the optimum μ and the selection parameter λ

```
means <- function(x,opt,sel) opt*(1-exp(-meanslist*sel)) + x*exp(-meanslist*sel);
```

The covariance matrix depends upon *t*, *s*, selection parameter λ and drift parameter σ.

```
covm <- function(sel,drift) (drift^2/(2*sel)) * exp(-sel*cmatrix) *
       (exp(2*sel*sdmatrix) - 1);
```

For a vector expression values corresponding the seven *Drosophila* species `data`, the density function of the OU model is given as the log likelihood of drawing `data[1]` from the equilibrium distribution, plus the log likelihood of drawing `data[2:7]` given `data[1]`. Here, the log likelihood is multiplied by –1 to aid in later numerical optimization. Additionally, the output is set to 10000 when either `sel` or `drift` goes below 0. This is to ensure the numerical optimization only returns positive values for `sel` and `drift`.

```
pdf <- function(param,data) {
       opt <- param[1]; sel <- param[2]; drift <- param[3];
       if ( sel>0 & drift>0 ) { -sum(
              dnorm(data[1],mean=opt,sd=drift/sqrt(2*sel),log=TRUE),
              dmvnorm(data[2:7],opt*(1-exp(-meanslist*sel)) + data[1]*exp(-
                     meanslist*sel),(drift^2/(2*sel)) * exp(-sel*cmatrix) *
                     (exp(2*sel*sdmatrix) - 1),log=TRUE)
       )}
       else { 10000 }
};
```

As an example, if opt = 0, sel = 100 and drift = 10, then the –log likelihood of observing the data is:

```
> pdf(c(0,100,10),expr)
[1] 8.37288
```

Estimation by maximum-likelihood

Numerical optimization is used get the maximum likelihood estimate of parameter values.

```
> optim(c(0,1,1),pdf,gr=NULL,expr)
$par
[1]   0.7882058 346.9209821   6.5069844
$value
[1] 0.1444493
```

So, for FBgn0039527, ML estimates μ at 0.79, λ at 346.9 and σ at 6.51. This gives as equilibrium variance of:

```
> 6.51^2/(2*346.92)
[1] 0.06108051
```

Testing significance of selection vs. neutrality

Want to compare the log likelihood of a model where μ, λ and σ are free variables (selection) to a model where the equilibrium variance is constrained to be 1 (neutrality).

The PDF for the neutral model as follows:

```
pdfn <- function(param,data) {
       opt <- param[1]; sel <- param[2]; sel <- (drift^2)/2
       if ( sel>0 & drift>0 ) { -sum(
              dnorm(data[1],mean=opt,sd=drift/sqrt(2*sel),log=TRUE),
              dmvnorm(data[2:7],opt*(1-exp(-meanslist*sel)) + data[1]*exp(-
                     meanslist*sel),(drift^2/(2*sel)) * exp(-sel*cmatrix) *
                     (exp(2*sel*sdmatrix) - 1),log=TRUE)
       )}
       else { 10000 }
};
```

This gives an ML estimate of:

```
> optim(c(0,1),pdfn,gr=NULL,expr)
$par
[1] 0.7313387 1.9256308
$value
[1] 4.589143
```

Because one free parameter differentiates these models, two times the difference between their log likelihoods is expected to be χ^2 distributed with one degree of freedom.

```
> 1-pchisq(2*(4.589143-0.1444493),1)
[1] 0.002868330
```

Thus, the p-value for the significance of selection on FBgn0011768 is 0.0029.