

MrAICM
written by C. R. Young
7/17/07

MrAICM.c is a utility written in the C programming language that applies the stabilized model discrimination measure AICM (with associated MCMC standard error) to Bayesian phylogenetic analyses. This measure is a posterior simulation-based analogue of the AIC model selection criteria (Akaike 1973). This method is described in:

*Raftery, A.E., M.A. Newton, J.M. Satagopan, P.N. Krivitsky (2007)
Estimating the Integrated Likelihood via Posterior Simulation Using the
Harmonic Mean Identity. In Bayesian Statistics 8, pp. 1–45. J.M.
Bernardo, M.J. Bayarri, J.O. Berger, A.P. Dawid, D. Heckerman, A.F.M.
Smith and M. West (Eds.), Oxford University Press.*

1. Method

Objective

We are interested in comparing various models (i.e., hypotheses) to determine the extent to which data lend statistical support the competing models. Because the integrated likelihood distribution describes the probability of the data under the model, we can use these distributions to measure statistical support for each model. One such method of comparing models using the integrated likelihood is the harmonic mean approximation of Bayes' factors. This method is implemented in the phylogenetic analysis program MrBayes. The harmonic mean approximation has been shown, however, to be numerically unstable by which we mean can have an infinite variance (e.g., Raftery *et al.* 2007). Here, we apply a numerically stable discrimination measure, the *Akaike information criterion-Monte Carlo* (AICM) (Raftery *et al.* 2007). AICM is an analogue of the *Akaike information criterion* (AIC) (Akaike 1973), which is defined as:

$$AIC = 2(L_{\max} - d),$$

where L_{\max} is the maximum likelihood of the data determined by finding point estimates of model parameters that lead to the maximum likelihood and d is the number of parameters in the model. Analogously, AICM is defined as:

$$AICM = 2(\hat{\ell}_{\max} - \hat{d}),$$

where $\hat{\ell}_{\max}$ is an *estimate* of the maximum likelihood of the data derived from posterior simulation and \hat{d} is an *estimate* of the effective number of parameters in the model, also derived from posterior simulation. Note that AICM is computed from the *marginal posterior distribution* of log-likelihood scores for the data (i.e., the integrated likelihood distribution).

The *integrated likelihood distribution* of a model is obtained by integrating over all of the parameters of a model to obtain the *marginal posterior distribution* of loglikelihood scores for the data (the *joint probability* of all of the data observations measured on the natural log scale). This integration can be performed numerically, as in

Bayesian phylogenetic analyses, by using Markov chain Monte Carlo (MCMC) methodology. During an MCMC run, T samples are drawn from the joint posterior distribution of the model parameters (as well as loglikelihood scores, L_t) in an identically distributed but not necessarily independent manner (i.e., sequential draws are *autocorrelated*). These samples are stored in the MCMC dataset (a matrix with one row for each simulation replication, t , and one column for each quantity of interest to be *monitored* in the sampling). Each column of the MCMC dataset contains draws from the marginal posterior distribution for that quantity (where we have integrated out the other parameters of the model), and can be summarized descriptively (e.g., means, standard deviations, and density traces). We can learn about the marginal posterior distributions of any parameter to arbitrary precision depending on the total number of samples, T .

By approximating the *integrated likelihood distribution* by a shifted gamma distribution, we are able to relate quantities that we can easily estimate from the marginal posterior distribution of log-likelihood scores (i.e., means and variances) to the difficult-to-obtain estimates that we need to calculate AICM (i.e., $\hat{\ell}_{\max}$ and \hat{d}). In the rest of this section, I describe the rationale behind AICM and expand on current results as follows:

1. Define the *shifted gamma distribution*.
2. Calculate the *moments* of the shifted gamma distribution analytically.
3. Map the integrated likelihood distribution to a shifted gamma distribution by relating the mean and variance of the *marginal posterior distribution* of log-likelihood scores to the mean and variance of the *shifted gamma distribution*.
4. Relate the *mean and variance* of the marginal posterior distribution of loglikelihood scores to the *parameters* of the shifted gamma distribution using *moment estimation* assuming a large sample size.
5. *Define AICM* under the large sample assumption in 4. We also describe the *MCMC sampling error* associated with our estimate of AICM, which is crucial to determine whether we have sampled enough from the marginal posterior distribution of loglikelihood scores to discriminate among models.
6. Describe a model weighting method that specifies the *relative statistical evidence* for each model that is examined with AICM.

1.1 The Shifted Gamma Distribution

We approximate the difference between the true maximum likelihood and independent and identically distributed (IID) draws from the posterior of loglikelihoods (i.e., L_t) using a gamma distribution:

$$X \equiv \Delta L = L_{\max} - L_t$$

$$X \sim \text{Gamma}(\alpha, \lambda^{-1})$$

where α is the *shape* parameter and $\lambda < 1$ (but close to one) is the *scale* parameter. (The symbol \equiv means *is equivalent to*, and the tilde means *is distributed as*.) In this context, $\alpha = d/2$, where d is the number of parameters underlying the model and X is an IID draw from the gamma distribution.

The probability density function (*pdf*) of X is:

$$f(x | \alpha, \lambda) = c(\alpha, \lambda)x^{(\alpha-1)} \exp(-x / \lambda) ,$$

where values of x from the gamma distribution are always greater than zero ($x > 0$). Integrating $f(x)$ gives:

$$c(\alpha, \lambda) \int_0^{\infty} x^{(\alpha-1)} \exp(-x / \lambda) dx = \Gamma(\alpha) \lambda^{\alpha} ,$$

where $\Gamma(\alpha)$ is the gamma function. $c(\alpha, \lambda) = \frac{1}{\Gamma(\alpha) \lambda^{\alpha}}$ is a normalization constant that is

required to make $\int_0^{\infty} f(x) = 1$, and thus a proper probability distribution. So we have the probability density function (*pdf*) of X is:

$$f(x) = \frac{x^{(\alpha-1)} \exp(-x / \lambda)}{\Gamma(\alpha) \lambda^{\alpha}} \quad (1)$$

Note: It is important to recognize that $f(x)$ is not the distribution of log-likelihood scores (as one would have in an MCMC dataset, for example); rather it is the distribution of differences (distances) of individual draws from the maximum possible log-likelihood of the model (see Fig. 1).

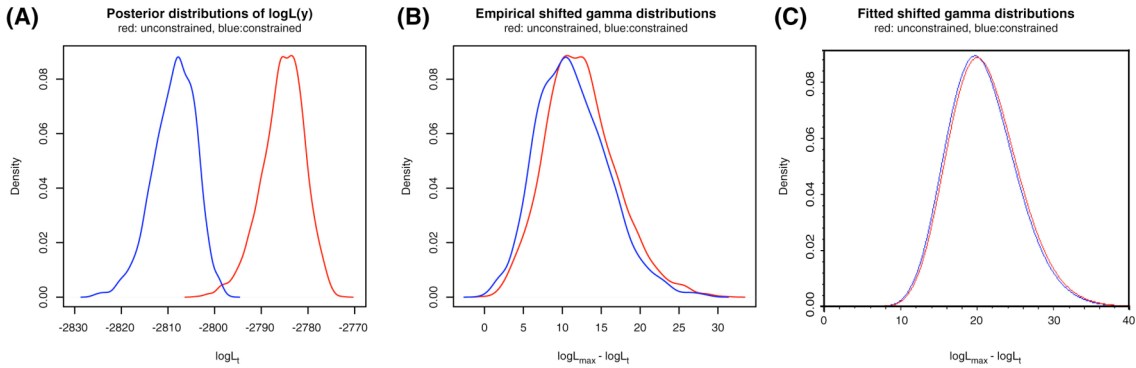


Figure 1. Log-likelihood and shifted gamma distributions from an unconstrained and constrained model. Data are phylogenetic sequences of 12 taxa, and the model constraint is monophyly of a group. (A) Posterior distributions of data log-likelihoods (L_i). (B) The empirical shifted gamma distribution computed from the raw data as the difference between the maximum *observed* logL and each logL in the MCMC dataset. (C) The shifted gamma distribution parameterized by *moments estimation* using the empirical data. Notice that the maximum *observed* logL is approximately 10 logL units from the estimated *maximum* logL. Also notice that the variances of the distributions are very similar (unconstrained $s_{\hat{\ell}}^2 = 21.0$, and constrained $s_{\hat{\ell}}^2 = 20.7$; unconstrained $\hat{d} = 41.9$, constrained $\hat{d} = 41.4$). Therefore, the penalty associated with relaxing the parameter constraint (about 0.6 log-likelihood units) is dwarfed by the difference in the likelihood of the data under the two models (about 20 log-likelihood units).

1.2 Calculating Means and Variances of Statistical Distributions

The expectation (or mean) of X is:

$$E(X) \equiv \int_0^{\infty} xf(x)dx \quad (2)$$

This integral is can be viewed as the continuum limit of a weighted average from a histogram. If x_i is the value of the bin and $h(x_i)$ is the height of the bin measured as a proportion of the total items in the histogram:

$$E(X) = \sum_{i=1}^n x_i h(x_i),$$

As the bin widths go to zero $h(x_i) \rightarrow f(x)dx$ and the discrete sum becomes an integral.

The variance is:

$$Var(X) \equiv \int_0^{\infty} (x - E(X))^2 f(x)dx \quad (3)$$

These are examples of *moments* of $f(x)$. In general the i^{th} moment of $f(x)$ with respect to ϕ is $M_i \equiv \int_0^{\infty} \phi^i f(x)dx$. For the expectation $\phi = x$, and for the variance $\phi = (x - E(X))^2$.

We use these definitions to compute the mean and variance of the shifted gamma distribution in terms of the parameters, α and λ . By substituting the function f into Equations 2 and 3 we find that the mean and the variance of the shifted gamma distribution (e.g., as stated in Raferty *et al.* 2007) are:

$$\begin{aligned} E(X) &= \alpha\lambda \\ Var(X) &= \alpha\lambda^2 \end{aligned} \quad (4)$$

See appendix 1 for MAPLE code to do the computation. These formulas allow us to estimate the parameters α and λ using approximations of $E(X)$ and $Var(X)$ computed from the data.

1.3 Mapping the Sample Moments to the Shifted Gamma Distribution

We can compute L_t from the MCMC data, but so far, our work has focused on the random variable $X = L_{max} - L_t$. We need to know how the mean and variance of our data L_t relate to the mean and variance of X . We use the fact that, for any real constants a and b , the following two properties of means and variances are true:

$$\begin{aligned} E(aX + b) &= aE(X) + b \\ \text{Var}(aX + b) &= a^2\text{Var}(X) \end{aligned} \tag{5}$$

Let's explore how the expectations and variances of the two distributions relate. For our MCMC data, the expectation is $E(L_t)$ and the variance is $\text{Var}(L_t)$. For the shifted gamma model, the expectation is $E(L_{\max}-L_t)$ and the variance is $\text{Var}(L_{\max}-L_t)$. Using the equations in (5):

$$\begin{aligned} E(-L_t + L_{\max}) &= -E(L_t) + L_{\max} \\ &= L_{\max} - E(L_t) \\ \text{Var}(-L_t + L_{\max}) &= -1^2\text{Var}(L_t) \\ &= \text{Var}(L_t) \end{aligned} \tag{6}$$

It seems that the only extra thing that we need to know is L_{\max} . Unfortunately, in MCMC applications, we are very unlikely to visit L_{\max} , and often the maximum log-likelihood in our MCMC dataset will be far from the true maximum value (see Fig. 1B and 1C). We will have to deal with this later. Remember that our ultimate goal is to be able to compute AICM, which also contains the pesky quantity L_{\max} (or rather a sample *estimate* of it: $\hat{\ell}_{\max}$).

Table 1. Notation used for MCMC sample estimates of loglikelihoods.

Estimate*	Meaning
$\hat{\ell}_{\max}$	Estimate of the maximum possible log-likelihood for the data: L_{\max}
$\bar{\ell}$	Expected value of the observed log-likelihood MCMC data: $E(L_t)$
s_{ℓ}^2	Variance of the observed log-likelihood MCMC data: $\text{Var}(L_t)$
\hat{d}	Effective number of parameters of the model

* notation from Raftery et al. (2007)

1.4 The Moment Estimator Assuming $\lambda = 1$

We can use *moments* (mean and variance of the sample) to estimate the parameters of a distribution. We have relations between the mean and variance of the gamma distribution and the parameters (Eqs. 4 and 6), so we could solve for the parameters, and say, use a sample mean and variance to estimate the parameters. We also will make an additional assumption before we proceed: since we know that as the amount of data (information) contributing to the loglikelihoods increases λ approaches 1, we make the *large-sample approximation* that

$$X \sim \text{Gamma}(\alpha, 1),$$

which gives us: $E(X) = \alpha$ and $\text{Var}(X) = \alpha$, by Equation 4 since we have assumed that $\lambda=1$.

Therefore:

$$\begin{aligned}
\hat{\alpha} &= E(X) = Var(X) \\
&= L_{\max} - E(L_t) \\
&= Var(L_t)
\end{aligned} \tag{7}$$

Remember that we do not know L_{\max} , but we can use the above relations to solve for it:

$$\begin{aligned}
L_{\max} - E(L_t) &= Var(L_t) \\
L_{\max} &= E(L_t) + Var(L_t)
\end{aligned} \tag{8}$$

Equation 8 tells us how to estimate $\hat{\ell}_{\max}$ from our MCMC data. Using the notation in Raferty et al. (2007) (see Table 1), $\hat{\ell}_{\max} = \bar{\ell} + s_{\ell}^2$, where $\bar{\ell} = E(L_t)$ and $s_{\ell}^2 = Var(L_t)$. We now need to know how to estimate \hat{d} . Raferty et al. (2007) arrive at an estimate of \hat{d} in two ways:

- 1) Recognizing that $X \sim Gamma(\alpha, \lambda^{-1})$ we see that $f(x)$ can be viewed as a scaled χ^2 distribution with $d = 2\alpha$ degrees of freedom.
- 2) The second comes from computing the integrated log-likelihood (i.e., $\log\pi(x)$, where x is the data) from $X \sim Gamma(\alpha, \lambda^{-1})$. This calculation results in the expression:

$$\log\pi(x) = L_{\max} + \alpha \log(1 - \lambda),$$

which is similar to the Bayesian Information Criterion (BIC) approximation to the log integrated likelihood:

$$\log\pi_{BIC}(x) = L_{\max} - \frac{d}{2} \log(n).$$

Since $\log\pi(y)$ and $\log\pi_{BIC}(y)$ converge (become the same) as the amount of data increases, the authors equate $\alpha = d/2$ and $-\log(1 - \lambda) = \log(n)$ in the two expressions. (**Note:** This is a weak analogy without point 1, since essentially the analogy says we have something of the form $ab = cd$. By saying $a = c$ and $b = d$, they are picking one solution out of an *infinite number* of possible solutions.)

In both of these cases, we have:

$$\hat{d} = 2\alpha = 2s_{\ell}^2. \tag{9}$$

1.5 Estimating AICM from the MCMC Data

The Akaike information criterion is defined as:

$$AIC = 2\ell_{\max} - 2d,$$

where d is the number of parameters in the model. Note here that $\hat{\ell}_{\max}$ is an *approximation* that has some associated error. To estimate AICM:

$$AICM = 2\hat{\ell}_{\max} - 2\hat{d}.$$

Since $\hat{\ell}_{\max} = \bar{\ell} + s_{\ell}^2$ and $\hat{d} = 2s_{\ell}^2$,

$$\begin{aligned} AICM &= 2(\bar{\ell} + s_{\ell}^2) - 4s_{\ell}^2 \\ &= 2(\bar{\ell} - s_{\ell}^2) \end{aligned}$$

We use the bottom expression to calculate AICM in this program. Note that Raftery et al. point out that AICM is *equivalent* to the definition of the *deviance information criterion* (DIC) in Gelman *et al.* (2003). An estimate of the Monte Carlo standard error (MCSE). Given B *approximately independent* MCMC draws,

$$MCSE_{AICM} = \sqrt{4\hat{d}/(2B) + 4\hat{d}(11\hat{d}/4 + 12)/B}, \quad (15)$$

which we report in the program.

Note: It is up to the user to ensure that the MCMC draws input into the program are *approximately independent* (not autocorrelated). This is achieved by *thinning* the MCMC dataset (i.e., by sub-sampling MCMC dataset every t simulation replications, with t chosen to reduce autocorrelation between successive replications so that they are approximately independent). Otherwise, equation 8 will be incorrect.

1.6 Computing the Akaike Weights from AICM

We are also interested in computing the *relative support* for different models that we compare. We can measure relative support by computing the Akaike weights, w_i , for each model i (Burnham & Anderson 2002). First, we must identify the best K-L model in the set R of competing models, which is the model with the maximum AICM, $AICM_{\max}$ (or equivalently the minimum AIC, since the signs of AIC and AICM are defined to be different). We will define the difference between the best model and other models that we wish to compare as:

$$\Delta AICM_i = AICM_{\max} - AICM_i.$$

Note that since AIC is on the log-probability scale, subtracting the two measures gives $\Delta AICM_i$ the interpretation of a *relative probability* on the raw probability scale. If we remove the scaling factor of 2 and exponentiate, then we recover this relative probability:

$$P_i = \exp\left(-\frac{1}{2}\Delta AICM_i\right).$$

The Akaike weights are defined as:

$$w_i = \frac{\exp\left(-\frac{1}{2}\Delta AICM_i\right)}{\sum_{j=1}^R \exp\left(-\frac{1}{2}\Delta AICM_j\right)}, \quad (16)$$

which we report in the program.

References

- Akaike, H. 1973. Information theory as an extension to the maximum likelihood principle. Pages 267-281 in Second international symposium on information theory (Petrov, B. N., and F Csaki, eds.) Akademiai Kiado, Budapest.
- Burnham, K. P. and D. R. Anderson, 2002. Model selection and multi model inference. A practical information-theoretic approach. Springer, New York.
- Raftery, A.E., M.A. Newton, J.M. Satagopan, P.N. Krivitsky (2007) Estimating the Integrated Likelihood via Posterior Simulation Using the Harmonic Mean Identity. In Bayesian Statistics 8, pp. 1–45. J.M. Bernardo, M J. Bayarri, J.O. Berger, A.P. Dawid, D. Heckerman, A.F.M. Smith and M. West (Eds.), Oxford University Press.

2. Compilation

To compile MrAICM from the source code, MrAICM.c, navigate to the directory that includes the source code and type:

```
gcc MrAICM.c -o MrAICM -lm
```

3. Program input

MrAICM requires several input files (present in the MrAICM directory). The first is the input file “files.txt” (can have any name). This file tells the program how many hypotheses to evaluate, the number and names of the *.p files in the directory that contain the MCMC samples for each hypothesis, and the burnin/thinning intervals to apply to each set of *.p files.

Format of “files.txt”:

```
<# of hypotheses>
<# of *.p files for H1> <burnin for H1> <thinning interval for H1>
...
<# of *.p files for Hn> <burnin for Hn> <thinning interval for Hn>

< filename of H1 >.run1.p
< filename of H1 >.run2.p
...

...

< filename of Hn >.run1.p
< filename of Hn >.run2.p
...

```

Example of files.txt:

```
2
2 1001 2
2 1001 2

sym_tentaxa_ONETree_AIC3x.run1.p
sym_tentaxa_ONETree_AIC3x.run2.p

sym_tentaxa_twotree_AIC3x.run1.p
sym_tentaxa_twotree_AIC3x.run2.p

```

In this example, there are two hypotheses to test. The next two lines (if there were three hypotheses, then there would be three lines here) specify the program parameters applied to each hypothesis. The MrBayes runs under the first hypothesis, `sym_tentaxa_ONETree_AIC3x`, produced two *.p files. The program will apply a burnin of 1001 to each *.p file, and the loglikelihood scores will be sub-sampled every two draws. The MrBayes runs under the second hypothesis, `sym_tentaxa_twotree_AIC3x`, produced two *.p files. The program will apply a burnin of 1001 to this set of files, and the loglikelihood scores will be subsampled every two draws. Following these parameters is a list of the *.p files in the order corresponding to the parameter specifications above. The remaining files that are required to run the program are the *.p files that are listed “files.txt.” These files must be present in the MrAICM directory.

4. Program output

To run the program, type:

```
./MrAICM files.txt
```

or to save the output to a file, use the redirect operator:

```
./MrAICM files.txt > output.txt
```

MrAICM reiterates the settings from the input file and reports several quantities of interest in the last few lines of the output (Table 2, Example output).

Table 2. Quantities reported by MrAICM.

Quantity	Definition
Model	Model number as ordered in the input file. Models are listed in the output according to Akaike weights with the first model having the largest weight.
Akaike Weight	Relative support for different models (Eq. 16)
AICM	$AICM = 2\hat{\ell}_{\max} - 2\hat{d}$ (Section 1.5)
SE_AICM	Monte Carlo standard error of the AICM estimate (Eq. 15)
d_hat	Estimate of the effective number of parameters, \hat{d} (Eq. 9)
E(logL)	Mean of the posterior distribution of loglikelihood scores, $\bar{\ell}$
Var(logL)	Variance of the posterior distribution of loglikelihood scores, s_{ℓ}^2

Example output:

```

Reading file list...
2 models to compare.
-----
Information for model 1:
-----
2 files for model 1.
Burnin (applied to each file) is 1001 for model 1.
Thinning interval (applied to each file) is 2 for model 1.

-----
Information for model 2:
-----
2 files for model 2.
Burnin (applied to each file) is 1001 for model 2.
Thinning interval (applied to each file) is 2 for model 2.

Files for model 1:
sym_tentaxa_ONETree_AIC3x.run1.p
sym_tentaxa_ONETree_AIC3x.run2.p

Files for model 2:
sym_tentaxa_twotree_AIC3x.run1.p
sym_tentaxa_twotree_AIC3x.run2.p

-----
Scanning model 1:
-----
Number of draws in MCMC file: 5000
Number of draws after thinning and burnin: 2000
Number of draws in MCMC file: 5000
Number of draws after thinning and burnin: 2000
Total MCMC draws in model 1 (after burnin and thinning) = 4000

-----
Scanning model 2:
-----
Number of draws in MCMC file: 5000
Number of draws after thinning and burnin: 2000
Number of draws in MCMC file: 5000
Number of draws after thinning and burnin: 2000
Total MCMC draws in model 2 (after burnin and thinning) = 4000

Model Akaike Weight AICM SE_AICM d_hat E(logL) Var(logL)
2 1.00000 52128.5 8.0 149.9 -25989.3 75.0
1 0.00000 52300.4 7.3 137.7 -26081.3 68.9

```

The burnin in this example is 1001. Note that this burnin is applied to *each* *.p file specified in the list for the model. The thinning interval (2 in this example) is applied after the burnin is removed. For example, model 1 includes two *.p files, each of which contain 5000 MCMC draws (note that the *.p files also contain the initial values as the first line). After removal of the first 1001 draws as a burnin, each file then contains 4000 draws. Thinning every two draws leaves 2000 draws in each of the two files. The remaining draws from both files are combined for a total of 4000 draws after burnin and thinning for model 1.

Of particular interest to users are the Akaike weights, which specify the relative support for the models that were evaluated. The models are listed in descending order of support and are numbered according to the order specified in the input file. The Monte Carlo standard error can be used to determine whether the differences observed between AICM for the models could be due to Monte Carlo sampling error.

Appendix 1.

Let's convince ourselves that equation 4 is correct by using MAPLE to compute the expectation and variance of $f(x)$ using equations 1, 2, and 3. We use MAPLE to do the integration. I will specify MAPLE input code in red and its output in blue. The command "restart" clears MAPLE's memory. The "assume()" command places bounds on the parameters (MAPLE is a symbolic algebra package, and will use those constraints to simplify matters when it can). Telling MAPLE that $\lambda < 1$ amounts to telling it that lambda is close to one. The third line inputs equation 1.

```
> restart;
> assume(alpha>0, lambda=1);
> f_X := (x,alpha,lambda) -> (x^(alpha-1)*exp(-x/lambda)) /
(GAMMA(alpha) * lambda^alpha);
```

$$f_X := (x, \alpha, \lambda) \rightarrow \frac{x^{(\alpha - 1)} \exp\left(-\frac{x}{\lambda}\right)}{\text{GAMMA}(\alpha) \lambda^\alpha}$$

Let's use equation 2 to find the mean of $f(x)$:

```
> Mean_X := simplify(int(x*f_X(x,alpha,lambda), x=0..infinity));
Mean_X :=  $\tilde{\lambda} \tilde{\alpha}$ 
```

Here, we find that we agree that $E(X) = \alpha\lambda$ (the tildes remind us that we made assumptions about the values of the parameters). Now let's use equation 3 to compute the variance of $f(x)$:

```
> Var_X := simplify(int((x-Mean_X)^2 * f_X(x,alpha,lambda),
x=0..infinity));
```

$$\text{Var}_X := \tilde{\lambda}^2 \tilde{\alpha}$$

Again, we find that we agree that $\text{Var}(X) = \alpha\lambda^2$.