



R Functions to Symbolically Compute the Central and Non-central Moments of the Multivariate Normal Distribution

Kem Phillips

Abstract

The central moments of the multivariate normal distribution are functions of its $n \times n$ variance-covariance matrix Σ . These moments can be expressed symbolically as linear combinations of products of powers of the elements of Σ . A formula for these moments derived by differentiating the characteristic function is developed. The formula requires searching integer matrices for matrices whose n successive row and column sums equal the n exponents of the moment. This formula is implemented in R, with R functions to display moments in \LaTeX and to evaluate moments at specified variance-covariance matrices are included.

This vignette also describes how the `symmoments` package has been augmented to calculate symbolic representations of non-central multivariate moments, and to calculate the expected value of such moments as well as the expected value of multivariate polynomials. In addition, it is shown that first non-central multivariate moments are in a one-to-one relationship to phylogenetic trees. Functions making this correspondence, as well as a function for the correspondence with matchings are provided.

Keywords: non-central moments, multivariate Normal distribution, symbolic computation, multivariate polynomials, phylogenetic trees, R, \LaTeX .

1. Introduction

The central moments of an n -dimensional random vector X are defined as

$$m_{k_1, \dots, k_n} = E[(X_1 - \mu_1)^{k_1} (X_2 - \mu_2)^{k_2} \cdots (X_n - \mu_n)^{k_n}], \quad (1)$$

where $E[\cdots]$ denotes expected value. Suppose that X is distributed according to the multivariate normal distribution with mean μ and variance-covariance matrix

$$\Sigma = (\sigma_{ij}) \quad (2)$$

where the variance terms are $\sigma_{ii}, i = 1, \dots, n$, the covariance terms are $\sigma_{ij}, i \neq j$, and by symmetry $\sigma_{ij} = \sigma_{ji}$. For the multivariate normal distribution the central moments are not functions of the mean vector μ , and depend only on the variance-covariance terms σ_{ij} .

Simple cases are familiar. Setting μ to 0,

$$m_2 = E[X_1^2] = \sigma_{1,1} \quad (3)$$

$$m_{1,2} = E[X_1 X_2] = \sigma_{1,2} \quad (4)$$

Slightly more complicated cases can be computed directly or by manipulating simple expressions obtained for moments of the form $E[X_1 \cdots X_n]$ (Wikipedia contributors 2009). For example,

$$m_{2,2} = E[X_1^2 X_2^2] = 2\sigma_{1,2}^2 + \sigma_{1,1}\sigma_{2,2} \quad (5)$$

$$m_{2,1,1} = E[X_1^2 X_2^1 X_3^1] = 2\sigma_{1,2}\sigma_{1,3} + \sigma_{1,1}\sigma_{2,3} \quad (6)$$

$$m_{1,1} = E[X_1^3 X_2^1] = 3\sigma_{1,1}\sigma_{1,2} \quad (7)$$

Although some higher order moments follow known patterns, most are much harder to determine by simple calculations.

We wish to compute the symbolic expression of any moment m_{k_1, \dots, k_n} in terms of the $n(n+1)$ symbols σ_{ij} . Note that if $\sum_{i=1}^n k_i$ is an odd integer, the moment is 0, so that only moments where this sum is even need be considered.

The multivariate normal distribution is fundamental to mathematical statistics, and its moments play a central role in statistical methodology. Various methods have been developed to numerically compute them (Muirhead 1982, p. 46) and (Anderson 1971, p. 49). Kan (2008) developed a formula (his Proposition 1) for the central moments as a repeated sum. He gives an excellent review of other formulas that have been developed, and cites Isserlis (1918) as deriving the first expression for the central moments. Muirhead (page 49) used the matrix derivatives of the multivariate normal distribution's characteristic function to derive a formula for multivariate cumulants. Tracy and Sultan (1993) also used matrix derivatives to derive an expression for the distribution's moments (their Theorem 2) based on a recurrence relationship of the derivatives. This article develops a new explicit formula for the moments starting with the derivatives of the characteristic function. The expression for the moments is based on a search algorithm over certain integer matrices. This formula has been translated into R functions that produce symbolic representations of moments in terms of the variance-covariance terms σ_{ij} . In addition, a formula for non-central moments is developed based on central moments. This formula is programmed, and is extended to the expected values of multivariate polynomials. Finally, it is shown that first central moments are equivalent to phylogenetic trees, and R functions making this correspondence are described.

The functions described here are based on Phillips (2010), and are available in the package **symmoments** implemented in the R system for statistical computing (R Development Core Team 2008). Both R itself and the **symmoments** package (as well as all other packages used in this paper) are available under the terms of the General Public License (GPL) from the Comprehensive R Archive Network (CRAN, <http://CRAN.R-project.org/>).

2. Development of the formula for central moments

The moments of any distribution can be represented by the derivatives of the distribution's characteristic function. The characteristic function of the multivariate normal distribution is (Muirhead 1982, p. 5, 49)

$$E[e^{it^\top X}] = e^{it^\top \mu - \frac{1}{2}t^\top \Sigma t} \quad (8)$$

where $t = (t_1, t_2, \dots, t_n)$. Within a constant, the moment is the k_1, \dots, k_n -order derivative of the characteristic function evaluated at $t = 0$:

$$m_{k_1, \dots, k_n} = i^{-\sum_{i=1}^n k_i} \frac{d^{\sum_{i=1}^n k_i}}{d^{k_1} t_1 d^{k_2} t_2 \dots d^{k_n} t_n} E[e^{it^\top X}] |_{t=0} \quad (9)$$

where i is the imaginary unit. Expanding the exponential into an infinite sum, this is

$$m_{k_1, \dots, k_n} = i^{-\sum_{i=1}^n k_i} \frac{d^{\sum_{i=1}^n k_i}}{d^{k_1} t_1 d^{k_2} t_2 \dots d^{k_n} t_n} \sum_{\ell=0}^{\infty} (it^\top \mu - \frac{1}{2}(t^\top \Sigma t))^\ell / \ell! |_{t=0} \quad (10)$$

Since we are to compute the central moment, we will set $\mu = 0$, so that the term $it^\top \mu$ will not appear:

$$m_{k_1, \dots, k_n} = i^{-\sum_{i=1}^n k_i} \frac{d^{\sum_{i=1}^n k_i}}{d^{k_1} t_1 d^{k_2} t_2 \dots d^{k_n} t_n} \sum_{\ell=0}^{\infty} (-\frac{1}{2}(t^\top \Sigma t))^\ell / \ell! |_{t=0} \quad (11)$$

Since $\sum_{i=1}^n k_i$ is even, the term $i^{-\sum_{i=1}^n k_i} = (-1)^{-\sum_{i=1}^n k_i/2}$. We note that the term $i^{-\sum_{i=1}^n k_i}$ will ultimately cancel with the negative in the infinite sum, and will be omitted for convenience in notation.

The expression in t is

$$t^\top \Sigma t = \sum_{ij} \sigma_{ij} t_i t_j \quad (12)$$

We need to find the coefficient of $t_1^{\ell_1} t_2^{\ell_2} \dots t_n^{\ell_n}$ in

$$(t^\top \Sigma t)^\ell = \left(\sum_{ij} \sigma_{ij} t_i t_j \right)^\ell = \sum_{ij} \sigma_{ij} t_i t_j \dots \sum_{ij} \sigma_{ij} t_i t_j \quad (13)$$

All products in the elements of the sum will occur. Any product will be obtained by choosing $\sigma_{ij} t_i t_j$ a certain number of times, say ℓ_{ij} . Since one term is chosen from each of ℓ terms, $\sum_{ij} \ell_{ij} = \ell$. Further, for any such matrix (ℓ_{ij}) , there will be a term, since it can be constructed by choosing $\sigma_{ij} t_i t_j$ from the first ℓ_{ij} terms, and so forth for each (ij) until ℓ is exhausted. For any (ℓ_{ij}) there are

$$\binom{\ell}{\ell_{11} \dots \ell_{nn}} \quad (14)$$

ways to choose the terms, where this is the multinomial coefficient. So,

$$\left(\sum_{ij} \sigma_{ij} t_i t_j \right)^\ell = \sum_{\{(\ell_{ij}) | \sum_{ij} \ell_{ij} = \ell\}} \binom{\ell}{\ell_{11} \dots \ell_{nn}} \prod_{ij} (\sigma_{ij} t_i t_j)^{\ell_{ij}} \quad (15)$$

$$= \sum_{\{(\ell_{ij}) \mid \sum_{ij} \ell_{ij} = \ell\}} \binom{\ell}{\ell_{11} \dots \ell_{nn}} \prod_{ij} \sigma_{ij}^{\ell_{ij}} \prod_{ij} (t_i t_j)^{\ell_{ij}} \quad (16)$$

For the moment, we distinguish between σ_{ij} and σ_{ji} as symbols. As a result, each $\prod_{ij} \sigma_{ij}^{\ell_{ij}}$ is unique as determined by unique (ℓ_{ij}) . However, $t_i t_j = t_j t_i$, so since each σ_{ij} is combined with two t 's, the total exponent in t is 2ℓ . That is, a term $t_1^{\ell_1} t_2^{\ell_2} \dots t_n^{\ell_n}$ must have $\sum_{i=1}^n \ell_i = 2\ell$. We need to determine the terms for which, for any (ℓ_1, \dots, ℓ_n) ,

$$\prod_{ij} (t_i t_j)^{\ell_{ij}} = t_1^{\ell_1} \dots t_n^{\ell_n} \quad (17)$$

We will get t_k in the product in the following mutually exclusive cases:

Condition	Exponent of t_k
$i = k, j \neq k$	1
$i \neq k, j = k$	1
$i = j = k$	2

(18)

So the exponent of t_k will be

$$\sum_{i=k, j \neq k} \ell_{ij} + \sum_{i \neq k, j=k} \ell_{ij} + 2\ell_{kk} \quad (19)$$

This sum is obtained by adding the sum of ℓ_{ij} across row k to the sum across column k , since the diagonal element k occurs in both sums. That is, we get

$$\sum_i \ell_{ik} + \sum_j \ell_{kj} = \sum_i (\ell_{ik} + \ell_{ki}) = \ell_k \quad (20)$$

We can now partition the set of (ℓ_{ij}) in Equation 16 according to these sums, that is, $\{\ell_k, k = 1 \dots n\}$. As stated before, the sum of the exponents, ℓ_k , must be 2ℓ .

$$\begin{aligned} & \left(\sum_{ij} \sigma_{ij} t_i t_j \right)^\ell = \\ & \sum_{\{(\ell_1, \dots, \ell_n) \mid \sum_k \ell_k = 2\ell\}} \sum_{\{(\ell_{11}, \dots, \ell_{nn}) \mid \sum_i (\ell_{ik} + \ell_{ki}) = \ell_k, k=1 \dots n\}} \left(\binom{\ell}{\ell_{11} \dots \ell_{nn}} \prod_{ij} \sigma_{ij}^{\ell_{ij}} \right) \prod_{i=1}^n t_i^{\ell_i} \quad (21) \end{aligned}$$

Since differentiation is distributive with respect to addition and multiplication by constants, the derivative of the product of t s can be determined from the derivatives of the individual terms:

$$\frac{d^{\sum_{i=1}^n k_i}}{d^{k_1} t_1 d^{k_2} t_2 \dots d^{k_n} t_n} \prod_{i=1}^n t_i^{\ell_i} = \prod_{i=1}^n \frac{d^{k_i}}{dt_i^{k_i}} t_i^{\ell_i} \quad (22)$$

$$= \prod_{i=1}^n I\{k_i \leq \ell_i\} \frac{\ell_i!}{(\ell_i - k_i)!} t_i^{\ell_i - k_i} \quad (23)$$

Thus,

$$\frac{d^{\sum_{i=1}^n k_i}}{d^{k_1} d^{k_2} \dots d^{k_n}} \left(\sum_{ij} \sigma_{ij} t_i t_j \right)^\ell = \sum_{\{(\ell_1, \dots, \ell_n) | \sum_i \ell_i = 2\ell\}} \sum_{\{(\ell_{11}, \ell_{12}, \dots, \ell_{nn}) | \sum_i (\ell_{ih} + \ell_{hi}) = \ell_i, h=1, \dots, n\}} \left(\binom{\ell}{\ell_{11} \dots \ell_{nn}} \prod_{ij} \sigma_{ij}^{\ell_{ij}} \right) \prod_{i=1}^n I\{k_i \leq \ell_i\} \frac{\ell_i!}{(\ell_i - k_i)!} t_i^{\ell_i - k_i} \quad (24)$$

Incorporating the constants from Equation 11, noting again that the negative signs will cancel, the full sum is

$$\sum_{\ell=0}^{\infty} \left(\frac{1}{2} \right)^\ell / \ell! \sum_{\{(\ell_1, \dots, \ell_n) | \sum_i \ell_i = 2\ell\}} \left(\sum_{\{(\ell_{11}, \ell_{12}, \dots, \ell_{nn}) | \sum_j (\ell_{hj} + \ell_{jh}) = \ell_i, h=1, \dots, n\}} \binom{\ell}{\ell_{11} \dots \ell_{nn}} \prod_{ij} \sigma_{ij}^{\ell_{ij}} \right) \prod_{i=1}^n I\{k_i \leq \ell_i\} \frac{\ell_i!}{(\ell_i - k_i)!} t_i^{\ell_i - k_i} \quad (25)$$

Setting $t = 0$, only terms with $\ell_i = k_i$ for all i will remain. Otherwise, the only ℓ in the infinite sum which occurs is for $\ell = \sum_{i=1}^n k_i/2$. So this reduces to

$$\left(\frac{1}{2} \right)^{\sum_{i=1}^n k_i/2} / \left(\sum_{i=1}^n k_i/2 \right)! \left(\sum_{\{(\ell_{11}, \dots, \ell_{nn}) | \sum_j (\ell_{hj} + \ell_{jh}) = k_i, h=1, \dots, n\}} \binom{\sum_{i=1}^n k_i/2}{\ell_{11} \dots \ell_{nn}} \prod_{ij} \sigma_{ij}^{\ell_{ij}} \right) \prod_{i=1}^n k_i! \quad (26)$$

Rearranging the terms, we have

$$m_{k_1, \dots, k_n} = C \sum_{\{(\ell_{11}, \ell_{12}, \dots, \ell_{nn}) | \sum_{j=1}^n (\ell_{jh} + \ell_{hj}) = k_n, h=1, \dots, n\}} \binom{\sum_{i=1}^n k_i/2}{\ell_{11} \dots \ell_{nn}} \prod_{ij} \sigma_{ij}^{\ell_{ij}} \quad (27)$$

where

$$C = \frac{1}{2^{\sum_{i=1}^n k_i/2}} \left(\prod_{i=1}^n k_i! \right) / \left(\sum_{i=1}^n k_i/2 \right)! \quad (28)$$

This formula shows that evaluating m_{k_1, \dots, k_n} symbolically requires enumerating all $n \times n$ -dimensional matrices of non-negative integers, (ℓ_{ij}) , which satisfy the condition

$$\sum_{j=1}^n (\ell_{ji} + \ell_{ij}) = k_i, \quad i = 1, \dots, n \quad (29)$$

Conceding now that the symbols σ_{ij} and σ_{ji} signify the same entity, we can search for (ℓ_{ij}) by looking only at terms $\prod_{ij} \sigma_{ij}^{\ell_{ij}}$ for which $i \leq j$. In fact, any other matrix for the term can be obtained by decrementing ℓ_{ij} and incrementing ℓ_{ji} by the same integer for one or more subscripts for which $i < j$. For any $\sigma_{ij}^{\ell_{ij}}$ in the term, this can be done in $\ell_{ij} + 1$ ways. So there

are a total of $\prod_{i < j} (\ell_{ij} + 1)$ transpositions. The multinomial coefficients derived above must be applied separately to each of these (ℓ_{ij}) matrices. Thus, the full coefficient for a matrix will include as a multiplier the sum of these coefficients over all of the $\prod_{i < j} (\ell_{ij} + 1)$ transposed matrices. Let Υ be the set of upper-triangular integer matrices, and, for any $(\ell_{ij}) \in \Upsilon$, let $\Lambda((\ell_{ij}))$ be the set of all integer matrices (h_{ij}) obtained by so transposing (ℓ_{ij}) . Then the sum above can be decomposed in terms of Υ and $\Lambda((\ell_{ij}))$ for each $(\ell_{ij}) \in \Upsilon$:

$$m_{k_1, \dots, k_n} = C \sum_{\{(\ell_{11}, \ell_{12}, \dots, \ell_{nn}) \in \Upsilon \mid \sum_{j=1}^n (\ell_{jh} + \ell_{hj}) = k_h, h=1, \dots, n\}} \sum_{\{(h_{ij}) \in \Lambda((\ell_{ij}))\}} \left(\frac{\sum_{i=1}^n k_i/2}{h_{11} \dots h_{nn}} \right) \prod_{ij} \sigma_{ij}^{h_{ij}} \quad (30)$$

But by symmetry, the products in (σ_{ij}) are the same for each member of $\Lambda((\ell_{ij}))$, specifically $\prod_{ij} \sigma_{ij}^{\ell_{ij}}$. So the final formula is

$$m_{k_1, \dots, k_n} = \left[\frac{1}{2} \sum_{i=1}^n k_i/2 \right] \left(\frac{\prod_{i=1}^n k_i!}{(\sum_{i=1}^n k_i/2)!} \right) \sum_{\{(\ell_{11}, \ell_{12}, \dots, \ell_{nn}) \in \Upsilon \mid \sum_{j=1}^n (\ell_{jh} + \ell_{hj}) = k_h, h=1, \dots, n\}} \left[\sum_{\{(h_{ij}) \in \Lambda((\ell_{ij}))\}} \left(\frac{\sum_{i=1}^n k_i/2}{h_{11} \dots h_{nn}} \right) \prod_{ij} \sigma_{ij}^{\ell_{ij}} \right] \quad (31)$$

3. Non-central moments

We now drop the condition that $\mu = 0$. That is, we want to calculate the symbolic representation of $E[X_1^{k_1} X_2^{k_2} \dots X_n^{k_n} \mid \mu, \Sigma]$, where $X \sim N(\mu, \Sigma)$. First, transform to $Y = X - \mu \sim N(0, \Sigma)$, so that $E[Y \mid \mu, \Sigma] = 0$. Then, since $X = Y + \mu$,

$$E[X_1^{k_1} X_2^{k_2} \dots X_n^{k_n} \mid \mu, \Sigma] = E[(Y_1 + \mu_1)^{k_1} (Y_2 + \mu_2)^{k_2} \dots (Y_n + \mu_n)^{k_n} \mid \mu, \Sigma] \quad (32)$$

For each term in the product we can choose Y_i l_i times and μ_i $k_i - l_i$ times, and this can be done in

$$\binom{k_i}{l_i} \quad (33)$$

ways. So, using the notation $k = (k_1, k_2, \dots, k_n)$, this is

$$(Y_1 + \mu_1)^{k_1} (Y_2 + \mu_2)^{k_2} \dots (Y_n + \mu_n)^{k_n} = \sum_{0 \leq l_i \leq k_i} \prod_{i=1}^n \binom{k_i}{l_i} \prod_{i=1}^n Y_i^{l_i} \prod_{i=1}^n \mu_i^{k_i - l_i} \quad (34)$$

Therefore, since $E[Y_1^{l_1} Y_2^{l_2} \dots Y_n^{l_n} \mid \mu, \Sigma] = E[Y_1^{l_1} Y_2^{l_2} \dots Y_n^{l_n} \mid \Sigma] = m_{l_1, \dots, l_n}$,

$$E[X_1^{k_1} X_2^{k_2} \dots X_n^{k_n} \mid \mu, \Sigma] = \sum_{0 \leq l_i \leq k_i} \left(\prod_{i=1}^n \binom{k_i}{l_i} \prod_{i=1}^n \mu_i^{k_i - l_i} \right) m_{l_1, \dots, l_n} \quad (35)$$

There are $\prod_{i=1}^n (k_i + 1)$ such moments. However, since the m_{l_1, \dots, l_n} are central moments, they are 0 for odd moments, and we need to consider only even moments, although k itself

does not have to be even. So, this can be written

$$E[X_1^{k_1} X_2^{k_2} \cdots X_n^{k_n} | \mu, \Sigma] = \sum_{0 \leq l \leq k, l \text{ even}} \left(\prod_{i=1}^n \binom{k_i}{l_i} \prod_{i=1}^n \mu_i^{k_i - l_i} \right) m_{l_1, \dots, l_n} \quad (36)$$

This formula involves the symbols $\prod_{i=1}^n \mu_i^{l_i}$, for $0 \leq l \leq k$. There are $\prod_{i=1}^n (k_i + 1)$ such "μ-products". As a general formula, all these terms are required, but in numerical calculations if any $\mu_i = 0$, all μ-products containing that μ_i will be 0.

Each m_{l_1, \dots, l_n} term in the sum has a unique σ -product (such as $\sigma_{1,4} \sigma_{2,4}^2 \sigma_{3,3}$), so that the μ-products, which are also unique, cannot be combined across σ -expressions. That is, the sum above is the best we can do.

Here is an example of a non-central moment:

$$\begin{aligned} E[X_1^1 X_2^2 X_3^3 | \mu, \Sigma] = & \\ & \mu_1 \mu_2^2 \mu_3^3 \\ & + 2 \mu_2 \mu_3^3 (\sigma_{1,2}) \\ & + \mu_1 \mu_3^3 (\sigma_{2,2}) \\ & + 3 \mu_2^2 \mu_3^2 (\sigma_{1,3}) \\ & + 6 \mu_1 \mu_2 \mu_3^2 (\sigma_{2,3}) \\ & + 3 \mu_3^2 (2\sigma_{1,2} \sigma_{2,3} + \sigma_{1,3} \sigma_{2,2}) \\ & + 3 \mu_1 \mu_2^2 \mu_3 (\sigma_{3,3}) \\ & + 6 \mu_2 \mu_3 (\sigma_{1,2} \sigma_{3,3} + 2\sigma_{1,3} \sigma_{2,3}) \\ & + 3 \mu_1 \mu_3 (\sigma_{2,2} \sigma_{3,3} + 2\sigma_{2,3}^2) \\ & + \mu_2^2 (3\sigma_{1,3} \sigma_{3,3}) \\ & + 2 \mu_1 \mu_2 (3\sigma_{2,3} \sigma_{3,3}) \\ & + (6\sigma_{1,2} \sigma_{2,3} \sigma_{3,3} + 3\sigma_{1,3} \sigma_{2,2} \sigma_{3,3} + 6\sigma_{1,3} \sigma_{2,3}^2) \end{aligned} \quad (37)$$

4. Expected Values of Multivariate Polynomials

By a multivariate polynomial, we mean a linear combination of products of powers of the components of X . That is, if K is a set of exponents $k = (k_1, \dots, k_r)$, and $\{\gamma_k | k \in K\}$ is a corresponding set of constants, a multivariate polynomial p is defined as

$$p(X_1, \dots, X_r) = \sum_{k \in K} \gamma_k X_1^{k_1} \cdots X_r^{k_r} \quad (38)$$

For example,

$$p(X_1, X_2, X_3) = 5 + 3X_1 X_2^2 + X_1^2 X_2^3 + X_3^4 \quad (39)$$

Multivariate polynomials are discussed by [Hankin \(2010\)](#) and [Kahle \(2014\)](#). Functions for the evaluation and algebra of multivariate polynomials is included in the `multipol` and `mpoly` packages.

A multivariate polynomial can be integrated against the multivariate normal distribution to give its expected value. Since these polynomials are linear combinations of products of powers of the components of X , their expectations are obtained easily from the corresponding non-central moments. So if $X \sim N(\mu, \Sigma)$, the expectation of the general multivariate polynomial above is

$$E[p(X_1, \dots, X_r) \mid \mu, \Sigma] = \sum_{k \in K} \gamma_k E[X_1^{k_1} \cdots X_r^{k_r} \mid \mu, \Sigma] \quad (40)$$

where the expectations are given in equation 36. Functions to compute these expectations are described below.

5. Multivariate moments, matchings, and phylogenetic trees

Phylogenetic trees are used to ascertain evolutionary relationships among species (Felsenstein 2004). A common model is the rooted, bifurcating tree. Diaconis and Holmes (1998) describe such a tree as “a binary rooted tree with n labeled leaves.” Felsenstein (Chapter 3) shows that the number of rooted, bifurcating trees for n species is

$$\frac{(2n - 3)!}{2^{n-2}(n - 2)!} \quad (41)$$

He also discusses the numbers of distinct shapes of trees and displays these shapes for $n = 2$ to $n = 6$ species. Diaconis and Holmes described how such phylogenetic trees are closely related to matchings.

Surprisingly, at least for computationally feasible values of n , for the first moment

$$E[X_1 \cdots X_{2(n-1)}] \quad (42)$$

the number of upper-triangular integer matrices that determine the moment is the same as the number of phylogenetic trees for n species. In fact, for any number of species there is a one-to-one correspondence between two seemingly disparate objects, phylogenetic trees and the symbolic first central moments of the multivariate normal distribution. The correspondence is established through matchings.

Diaconis and Holmes define a perfect matching on $2n$ points as a “pairing of the points into n groups of two.” We show that there is a one-to-one correspondence, or bijection, between the components of a representation of a multivariate normal first moment with $2n$ components and matchings on $2n$ points.

For any n , let $\Lambda(2n)$ be the collection of $2n \times 2n$ -dimensional matrices associated with the symbolic first moment of the $2n$ -dimensional multivariate normal distribution. Let $\Gamma(2n)$ be the set of perfect matchings on a set Δ of $2n$ points.

First suppose that $L = (l_{ij}) \in \Lambda(2n)$. Then criterion 29 becomes

$$\sum_{j=1}^{2n} (l_{ji} + l_{ij}) = 1, \quad i = 1, \dots, 2n \quad (43)$$

That is, there is exactly one $l_{ij} = 1$ in the union of the cell in row k and column k for each $1 \leq k \leq 2n$. The sum of these sums over i is twice the total count of cells with 1s, and is obviously $2n$. Therefore, the number of cells containing 1 is n .

Let $\gamma = \{(i, j) \mid l_{i,j} = 1, i = 1, \dots, 2n, j = 1, \dots, 2n\}$, that is, the set of cells containing 1. By criterion 43, for any pair $(i, j) \in \Gamma$, $i \neq j$ since the diagonal terms in the sum occur twice, so that if $i = j$ we would have $k_j \geq 2$. Furthermore, if any two pairs (i_1, j_1) and (i_2, j_2) were to share a common index, say $i_1 = j_2 = h$, then the union of row h and column h would have at least two cells containing 1, leading to $k_h \geq 2$. Therefore, the indices in these cells are unique, and since there are n cells containing 1, there are $2n$ distinct indices among these cells, namely $1, \dots, 2n$. It is clear that γ is a perfect matching on this set of indices. So for any $L \in \Lambda(2n)$ there corresponds a perfect matching γ on $1, \dots, 2n$.

Now suppose that $\gamma \in \Gamma(2n)$. For specificity, assume that the points in γ are numbered $1, \dots, 2n$. Order the pairs in γ lexicographically by the second components so that for any pair (i, j) , $i < j$. Let $L = (l_{i,j})$ be a $2n \times 2n$ -dimensional matrix of 0s with the rows and columns labeled $1, \dots, 2n$. Set any cell in L corresponding to a pair in γ to 1. Since any point occurs in one and only one pair, for any index i there is one and only one cell in row i and column i with a 1 in it. Every point is in the matching, so for each i , there must be a 1 in one cell of row i and column i . So for any $\gamma \in \Gamma(2n)$ there is an $L \in \Lambda(2n)$.

Invoking the algorithm projecting $\Lambda(2n)$ into $\Gamma(2n)$, the L computed above will be mapped back into the matching just defined. That is, these two procedures are inverses.

Paradis describes various formats for encoding trees and R functions to transform one to another. Functions are described below to transform between an **ape** matching object or Newick format to a moment object. Using these functions and the functions `read.tree`, `write.tree`, `as.matching`, and `as.phylo.matching` from the **ape** package, trees can be transformed among the **moment**, **matching**, **phylo**, and **Newick** formats. Table 1 shows the functions used for each transformation. Empty cells can be reached by obvious paths.

From	To			
	<u>Newick</u>	<u>Phylo</u>	<u>Matching</u>	<u>Moment</u>
Newick	—————	<code>read.tree</code>		<code>toMoment</code>
Phylo	<code>write.tree</code>	—————	<code>as.Matching</code>	
Matching		<code>as.phylo.matching</code>	—————	<code>toMoment</code>
Moment	<code>toNewick</code>		<code>toMatching</code>	—————

Table 1: Functions for converting between tree formats

6. Discussion

Formula 31 was implemented in R with a recursive function that determines the set of upper-triangular integer matrices that satisfy Criterion 29. A second function calculates their associated coefficients. Additional functions were written to create L^AT_EX (L^AT_EX3 Project Team 2009) code to display the moments symbolically, and to calculate the moments for specified variance-covariance matrices, and these have been augmented with functions for non-central moments.

The potential for complexity in these computations is seen from the results in Table 2. In this table, n is the dimension of the multivariate vector and $\#(\sigma_{ij})$ is the number of distinct elements in the variance-covariance matrix, $N = \frac{n(n+1)}{2}$. *Size* is measured by two values,

M and r . M is the total of the exponents of the moment, $M = \sum_{i=1}^n k_i$. The value r is the number of terms for a moment $E[X_1^1, \dots, X_n^1]$ with all exponents equal to 1, which is $(2M-1)!/(2^{M-1}(M-1)!)$ (Wikipedia contributors 2009). *Example* is a moment of the given *Size*. *Potential Terms* is a maximum for the number of (ℓ_{ij}) matrices to be checked for this example, determined as the product of $1 + \max(k_i, k_j)$ over $i \neq j$ times the product of $1 + \lceil k_i/2 \rceil$ over i , where $\lceil \cdot \rceil$ denotes truncation. The last column, *# Terms*, is the actual number of terms in the moment as determined by the functions.

It is clear that computation of high-order moments will be very intensive. The equivalence of first moments of dimension $2(n-1)$ and phylogenetic trees for n species has been shown above. Note that it would be possible that the L -representation of symbolic multivariate normal moments is sufficient but not necessary to determine the moment. However, the L representation of a multivariate normal moment can be determined by inspection from the σ terms, so that the symbolic moment and its L representation are, in fact, equivalent.

Felsenstein (2004, page 61) notes that finding the best tree using maximum parsimony is NP-hard, as shown by Foulds and Graham (1982). Felsenstein also notes that the Fitch algorithm for finding the number of changes in a given bifurcating tree is efficient (Felsenstein 2004, page 11). This implies that finding all trees is itself NP-hard, and that therefore computing first multivariate moments is NP-hard. Note that for any even n and any integer r , the moment $E[X_1^r X_2^r \dots X_n^r]$ has L components whose terms are L matrices from the representation of $E[X_1^1 X_2^1 \dots X_n^1]$ multiplied by r . So the representation of this first moment can be obtained from inspection of the higher moment, implying that computing these higher moments is also NP-hard.

Finally, Criterion 29 might arise in other contexts, such as networks (Stergiou and Siganos 1996). For example, suppose that there are n airports and airport i can accommodate k_i arrivals or departures on a day, where a plane may take off and land at the same airport. This network is illustrated in Figure 1. The problem is to determine the set of flights between airports that totally expend the capacities, k_i , of all airports. For this problem, ℓ_{ij} represents the number of flights from airport i to airport j , and ℓ_{ii} is the number of flights which start and end at airport i . The set of flights is the set satisfying Criterion 29.

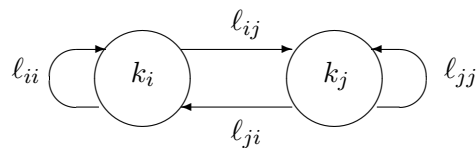


Figure 1: Airport Capacity Network

7. R functions

7.1. R functions for computing central moments

The following R functions calculate expressions and values for central and non-central moments.

In these R functions, upper-triangular matrices (ℓ_{ij}) for n dimensions are represented as vectors of length $n(n+1)$, with row 1 followed by row 2, etc. For example, for $n = 2$,

n	$\#(\sigma_{ij})$	M	Size	Example	Potential Terms	# Terms
	N		r	$E[X_1^{k_1} \dots X_n^{k_n}]$		
2	3	2	1	$E[X_1 X_2]$	2	1
2	3	6	15	$E[X_1^3 X_2^3]$	16	2
2	3	20	654,729,075	$E[X_1^{10} X_2^{10}]$	396	6
4	10	8	105	$E[X_1^2 X_2^2 X_3^2 X_4^2]$	11,664	17
4	10	12	10,395	$E[X_1 X_2 X_3 X_4 X_4]$	225,000	27
4	10	20	654,729,075	$E[X_1^5 X_2^5 X_3^5 X_4^5]$	3,779,136	306
6	21	12	10,395	$E[X_1^2 X_2^2 X_3^2 X_4^2 X_5^2 X_6^2]$	918,330,048	388
6	21	18	34,459,425	$E[X_1 X_2 X_3 X_4 X_5 X_6 X_6]$	1.265625×10^{12}	2,082
8	36	16	2,027,025	$E[X_1^2 X_2^2 X_3^2 X_4^2 X_5^2 X_6^2 X_7^2 X_8^2]$	5.856459×10^{15}	18,155
8	36	24	316,234,143,225	$E[X_1^3 X_2^3 X_3^3 X_4^3 X_5^3 X_6^3 X_7^3 X_8^3]$	1.844674×10^{19}	$\geq 287,800^*$
9	45	18	34,459,425	$E[X_1^2 X_2^2 X_3^2 X_4^2 X_5^2 X_6^2 X_7^2 X_8^2 X_9^2]$	7.68484×10^{19}	6,763,895

Table 2: Examples of sizes of moment problem (*: The function aborted after 2 days due a space allocation problem.)

(ℓ_{ij}) is represented as $(\ell_{11}, \ell_{12}, \ell_{22})$. Each such matrix represents the exponents for a single product of σ_{ijs} . For example, $(1, 2, 0)$ represents $\sigma_{11}^1 \sigma_{12}^2 \sigma_{22}^0 = \sigma_{11} \sigma_{12}^2$. The representations are accumulated and stored internally, raising the possibility of space allocation problems as encountered in the second to last example in Table 2. This problem could be alleviated by saving the representation matrix to a file instead.

The function `multmoments` searches the integer matrices for those satisfying Criterion 29. This is a recursive function which implements a branch-and-bound algorithm. The function `multmoments` is called by `callmultmoments`. This function initializes variables, determines the coefficients of the terms from the upper-triangular representations, and returns a list consisting of the original moment vector, the set of representations, and the corresponding set of coefficients. This list is set to class `moment`. The `moment` class has three methods: `print`, `toLatex`, `evaluate`, and `simulate`.

The `print` method prints a moment object, usually the output of `callmultmoments`, showing a mathematical representation of the moment, followed by the rows of the representation with the corresponding coefficient attached.

The `toLatex` method uses a moment object, usually the output of `callmultmoments`, to determine the L^AT_EX code for the moment sorted lexicographically. Note that it inserts double backslashes where L^AT_EX requires a backslash; these can be reset to single backslashes by writing the output to a file using the R function `writeLines` (**base** package), as illustrated below.

The `evaluate` method determines the value of a `moment` object for a specified variance-covariance matrix Σ , which must be represented as an upper-triangular matrix in vector form.

The `simulate` method uses Monte Carlo integration [Rizzo \(2008\)](#) to numerically approximate a `moment` object for a specified mean and variance-covariance matrix Σ (represented as a square matrix in vector form), with a specified number of random samples. Note that `simulate` uses only the moment definition, not the representation, so can be used with any moment in vector notation by converting the vector to a moment object as shown below. The `simulate` method uses the `rmvnorm` function from the **mvtnorm** package ([Genz, Bretz, with contributions by Tetsuhisa Miwa, Mi, Leisch, and Scheipl \(2008\)](#)).

7.2. R functions for computing non-central moments

The `toLatex_noncentral` function produces a Latex expression for a noncentral moment. This function uses central moments in the environment specified in the function call. The default environment is `symmoments`.

The `evaluate_noncentral` function evaluates a noncentral moment with specified mean and covariance matrix. This function uses central moments in the environment specified in the function call. The default environment is `symmoments`.

The `evaluate_expected.polynomial` function evaluates a multivariate polynomial, assuming a specified non-central multivariate distribution. This is done by calling `evaluate_noncentral` for each term in the polynomial. The polynomial is expressed as a `multipol` object, an `mpoly` object, or defined by a list giving the exponents in each term and its coefficient. The central moment objects required for this computation must be computed before invoking this function. The central moments must reside in the environment specified in the function call. The

default environment is `symmoments`.

The `convert.multipol` function converts between `multipol` objects and lists representing the same multivariate polynomial.

The `convert.mpoly` function converts between `mpoly` objects and lists representing the same multivariate polynomial. Note that `convert.multipol` and `convert.mpoly` can be composed to transform between these two types of objects. However, the names of the variables will not be preserved.

The `tounsorted` function produces an unsorted central moment object from a sorted object of class "moment". Unsorted moments are those with exponents not in numeric order, e.g., $m_{3,1,2}$.

The `make.all.moments` function creates all central moment objects of a specified or smaller size. That is, if $c(k_1, \dots, k_n)$ is input, all moments m_{l_1, \dots, l_n} with $(l_1, \dots, l_n) \leq (k_1, \dots, k_n)$ are produced. The moments are placed in the `symmoments` environment. If this environment does not exist, the user is prompted to create it. Moments that exist in the `symmoments` environment are not recreated. Unsorted moments, those with exponents are not in numeric order, are created using the `tounsorted` function to transform from the sorted moment. If the sorted moment does not exist, it is created. Moments of lower dimension are not created; for example, if $c(2, 4)$ is input, m_{20} is created, but m_2 is not.

As a convention, moments are named $m_{ij..l}$, e.g., $m136$. If any exponent is greater than 9, lower case letters and then upper case letters are used. For example, $m3bA$ is the name of the moment $m_{3,11,36}$. The function `callmultmoments` theoretically accommodates larger exponents, but the largest exponent allowed by this scheme is 61. If an object with a name of this form exists but is not an object of class "moment", it is replaced (overwritten) by the moment object.

The `integrate.polynomial` function integrates a multivariate polynomial against a specified non-central multivariate distribution using ordinary integration by invoking the `adaptIntegrate` function from the `cubature` package. This function was written mainly for checking.

7.3. R functions for correspondence with phylogenetic trees

Three R functions were written to implement the correspondences among the moment, matching, and Newick formats.

The `toMoment` function converts a tree to moment format. Its input is a tree in Newick format (Felsenstein 1990) or a `matching` object from the `ape` package (Paradis, Claude, and Strimmer 2004). `toMoment` outputs an object of class `L-matrix`. This class consists of 5 components as defined in the function description, and includes the L matrix. Note that the `symmoments` package defines the `moment` class, which encompasses all L -matrices for symbolic multivariate normal moments.

The `toNewick` function converts a tree in moment format to Newick format. The input can be an `L-matrix` object, a square L matrix, or an L matrix in reduced upper-triangular (vector) form. The `toNewick` function sets its list output to class `L-Newick`, which has 5 components, including the tree in Newick format.

The `toMatching` function converts a tree in moment format to an `ape matching` object. The input can be an `L-matrix` object, a square L matrix, or an L matrix in reduced upper-triangular (vector) form. `toMatching` sets its list output to class `matching` to conform to the

definition in the `ape` package.

7.4. Examples of computing central moments

These examples of the functions for central moments use the following moment:

$$\begin{aligned}
 E[X_1^1 X_2^2 X_3^3 X_4^4] = & \\
 & 18\sigma_{1,2}\sigma_{2,3}\sigma_{3,3}\sigma_{4,4}^2 + 72\sigma_{1,2}\sigma_{2,3}\sigma_{3,4}^2\sigma_{4,4} + 72\sigma_{1,2}\sigma_{2,4}\sigma_{3,3}\sigma_{3,4}\sigma_{4,4} + 48\sigma_{1,2}\sigma_{2,4}\sigma_{3,4}^3 + \\
 & 9\sigma_{1,3}\sigma_{2,2}\sigma_{3,3}\sigma_{4,4}^2 + 36\sigma_{1,3}\sigma_{2,2}\sigma_{3,4}^2\sigma_{4,4} + 18\sigma_{1,3}\sigma_{2,3}^2\sigma_{4,4}^2 + 144\sigma_{1,3}\sigma_{2,3}\sigma_{2,4}\sigma_{3,4}\sigma_{4,4} + \\
 & 36\sigma_{1,3}\sigma_{2,4}^2\sigma_{3,3}\sigma_{4,4} + 72\sigma_{1,3}\sigma_{2,4}^2\sigma_{3,4}^2 + 36\sigma_{1,4}\sigma_{2,2}\sigma_{3,3}\sigma_{3,4}\sigma_{4,4} + 24\sigma_{1,4}\sigma_{2,2}\sigma_{3,4}^3 + \\
 & 72\sigma_{1,4}\sigma_{2,3}^2\sigma_{3,4}\sigma_{4,4} + 72\sigma_{1,4}\sigma_{2,3}\sigma_{2,4}\sigma_{3,3}\sigma_{4,4} + 144\sigma_{1,4}\sigma_{2,3}\sigma_{2,4}\sigma_{3,4}^2 + 72\sigma_{1,4}\sigma_{2,4}^2\sigma_{3,3}\sigma_{3,4}
 \end{aligned} \tag{44}$$

The use of the `toLatex` and `evaluate` methods and `writeLines` is also illustrated. The file created by `writeLines` can be included in a \LaTeX document using the `\input` command, or can be included in Sweave as done here.

Compute the representation of a central moment (callmultmoments)

The following code calculates a central moment and shows the three components, **moment**, **representation**, and **coefficients**.

```
> m1234 <- callmultmoments(c(1,2,3,4))
> unclass(m1234)
```

\$moment
[1] 1 2 3 4

\$representation

	S(1,1)	S(1,2)	S(1,3)	S(1,4)	S(2,2)	S(2,3)	S(2,4)	S(3,3)	S(3,4)	S(4,4)
1	0	0	0	1	0	0	2	1	1	0
2	0	0	0	1	0	1	1	0	2	0
3	0	0	0	1	0	1	1	1	0	1
4	0	0	0	1	0	2	0	0	1	1
5	0	0	0	1	1	0	0	0	3	0
6	0	0	0	1	1	0	0	1	1	1
7	0	0	1	0	0	0	2	0	2	0
8	0	0	1	0	0	0	2	1	0	1
9	0	0	1	0	0	1	1	0	1	1
10	0	0	1	0	0	2	0	0	0	2
11	0	0	1	0	1	0	0	0	2	1
12	0	0	1	0	1	0	0	1	0	2
13	0	1	0	0	0	0	1	0	3	0
14	0	1	0	0	0	0	1	1	1	1
15	0	1	0	0	0	1	0	0	2	1
16	0	1	0	0	0	1	0	1	0	2

\$coefficients

rep 1	rep 2	rep 3	rep 4	rep 5	rep 6	rep 7	rep 8	rep 9	rep 10	rep 11
72	144	72	72	24	36	72	36	144	18	36
rep 12	rep 13	rep 14	rep 15	rep 16						
9	48	72	72	18						

Print a representation of a central moment (print method)

The following shows the result of using the `print` method with the moment in Equation 44.

```
> m1234
```

```
E[ X1^1 X2^2 X3^3 X4^4 ]:
```

	coef	S(1,1)	S(1,2)	S(1,3)	S(1,4)	S(2,2)	S(2,3)	S(2,4)	S(3,3)	S(3,4)	S(4,4)
1	72	0	0	0	1	0	0	2	1	1	0
2	144	0	0	0	1	0	1	1	0	2	0
3	72	0	0	0	1	0	1	1	1	0	1
4	72	0	0	0	1	0	2	0	0	1	1
5	24	0	0	0	1	1	0	0	0	3	0
6	36	0	0	0	1	1	0	0	1	1	1
7	72	0	0	1	0	0	0	2	0	2	0
8	36	0	0	1	0	0	0	2	1	0	1
9	144	0	0	1	0	0	1	1	0	1	1
10	18	0	0	1	0	0	2	0	0	0	2
11	36	0	0	1	0	1	0	0	0	2	1
12	9	0	0	1	0	1	0	0	1	0	2
13	48	0	1	0	0	0	0	1	0	3	0
14	72	0	1	0	0	0	0	1	1	1	1
15	72	0	1	0	0	0	1	0	0	2	1
16	18	0	1	0	0	0	1	0	1	0	2

Compute the \LaTeX representation of a central moment (toLatex method)

The following shows the computation of the representation of the central moment in Equation 44.

```
> toLatex(m1234)

[1] "E[X_{1}^{1}X_{2}^{2}X_{3}^{3}X_{4}^{4}]=\\\\"
[2] "18\\sigma_{1,2}\\sigma_{2,3}\\sigma_{3,3}\\sigma_{4,4}^{2}+"
[3] "72\\sigma_{1,2}\\sigma_{2,3}\\sigma_{3,4}^{2}\\sigma_{4,4}+"
[4] "72\\sigma_{1,2}\\sigma_{2,4}\\sigma_{3,3}\\sigma_{3,4}\\sigma_{4,4}+"
[5] "48\\sigma_{1,2}\\sigma_{2,4}\\sigma_{3,4}^{3}+\\\\"
[6] "9\\sigma_{1,3}\\sigma_{2,2}\\sigma_{3,3}\\sigma_{4,4}^{2}+"
[7] "36\\sigma_{1,3}\\sigma_{2,2}\\sigma_{3,4}^{2}\\sigma_{4,4}+"
[8] "18\\sigma_{1,3}\\sigma_{2,3}^{2}\\sigma_{4,4}^{2}+"
[9] "144\\sigma_{1,3}\\sigma_{2,3}\\sigma_{2,4}\\sigma_{3,4}\\sigma_{4,4}+\\\\"
[10] "36\\sigma_{1,3}\\sigma_{2,4}^{2}\\sigma_{3,3}\\sigma_{4,4}+"
[11] "72\\sigma_{1,3}\\sigma_{2,4}^{2}\\sigma_{3,4}^{2}+"
[12] "36\\sigma_{1,4}\\sigma_{2,2}\\sigma_{3,3}\\sigma_{3,4}\\sigma_{4,4}+"
[13] "24\\sigma_{1,4}\\sigma_{2,2}\\sigma_{3,4}^{3}+\\\\"
[14] "72\\sigma_{1,4}\\sigma_{2,3}^{2}\\sigma_{3,4}\\sigma_{4,4}+"
[15] "72\\sigma_{1,4}\\sigma_{2,3}\\sigma_{2,4}\\sigma_{3,3}\\sigma_{4,4}+"
[16] "144\\sigma_{1,4}\\sigma_{2,3}\\sigma_{2,4}\\sigma_{3,4}^{2}+"
[17] "72\\sigma_{1,4}\\sigma_{2,4}^{2}\\sigma_{3,3}\\sigma_{3,4}+\\\\"
```

The \LaTeX representation can be written to a file using the `writeLines` function as follows:

```
> writeLines(toLatex(m1234), "yourfilename")
```

Compute a value of a central moment (evaluate method)

The code below evaluates the moment at the following variance-covariance matrix:

	[,1]	[,2]	[,3]	[,4]
[1,]	4	2	1	1
[2,]	2	3	1	1
[3,]	1	1	2	1
[4,]	1	1	1	2

```
> evaluate(m1234, c(4, 2, 1, 1, 3, 1, 1, 2, 1, 2))
```

```
[1] 3480
```

Estimate a central moment using simulation (simulate method)

The value of $E[X_1^1 X_2^2 X_3^3 X_4^4]$ when X has a normal distribution with mean $\mu = (1, 2, 0, 3)$ and the same variance-covariance matrix as above could be estimated using `simulate` with 1000 random samples. More samples may be required to obtain an accurate estimate.

```
> #simulate(m1234,1000,NULL, c(1,2,0,3),c(4,2,1,1,2,3,1,1,1,1,2,1,1,1,1,2))
```

7.5. Examples of computing non-central moments

Note that several of these examples would require moments of various sizes exist, so have not been run. These moments can be created using the `make.all.moments` function, or directly using the `callmultmoments` function.

Calculate symbolic representation of a non-central moment (not run)

```
> # as.matrix(toLatex_noncentral(c(1,3)))
```

Create all 2-dimensional moment objects with exponents up to 3 (not run)

Note that this will overwrite objects in the `symmoments` environment with names of the form `mxx`, where $0 \leq x \leq 3$

```
> # make.all.moments(c(3,3))
```

Evaluate a non-central moment at a specified mean and covariance matrix (not run)

Note that this would require moments to exist of order up to `c(1,3)`.

```
> # evaluate_noncentral(c(1,3),c(1,2),c(1,0,1))
```

Create an mpoly object (not run)

```
> # t0 <- mpoly::mpoly(list(c(coef=3,x1=2),c(coef=2,x1=1,x2=3),
> #                               c(coef=-4,z=2),c(coef=1,x1=1,x2=2,z=1)))
> # print(t0)
```

Convert an mpoly object to a moment object (not run)

```
> # t1 <<- convert.mpoly(t0)
> # t1
```

Convert a moment object to a multipol object (not run)

```
> # t2 <- convert.multipol(t1)
> # t2
```

Convert from multipol back to mpoly through moment (not run)

```
> # print(mpoly::mpoly(convert.mpoly(convert.multipol(t2))))
```

Evaluate the expected value of a multivariate polynomial (not run)

See the Latex example in the text. This requires moments to exist of order up to $c(1, 2, 3)$.

```
> # evaluate_expected.polynomial(t0, c(1, 2, 3), c(1, 0, 0, 1, 0, 1))
```

7.6. Examples of correspondence between first multivariate moments and phylogenetic trees

The following examples convert a tree in Newick format to trees in phylo, matching, and moment format.

Create a Newick representation of a tree

```
> exam.Newick <- "((a,b),c),d);"
```

Convert to phylo format

```
> library(ape)
> exam.phylo <- read.tree(text=exam.Newick)
> exam.phylo
```

Phylogenetic tree with 4 tips and 3 internal nodes.

Tip labels:
a, b, c, d

Rooted; no branch lengths.

Convert to matching format

```
> exam.matching <- as.matching(exam.phylo)
> exam.matching
```

```
$matching
      [,1] [,2]
[1,]    1    2
[2,]    3    5
[3,]    4    6
```

```
$tip.label
[1] "a" "b" "c" "d"
```

```
attr("class")
[1] "matching"
```

Convert to L-matrix format

```
> exam.L.matrix <- toMoment(exam.matching)
> exam.L.matrix
```

```
$L
      [,1] [,2] [,3] [,4] [,5] [,6]
[1,]    0    1    0    0    0    0
[2,]    0    0    0    0    0    0
[3,]    0    0    0    0    1    0
[4,]    0    0    0    0    0    1
[5,]    0    0    0    0    0    0
[6,]    0    0    0    0    0    0
```

```
$L.ut
[1] 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0
```

```
$Newick
[1] "(((a,b),c),d);"
```

```
$tip.label
[1] "a" "b" "c" "d"
```

```
$tip.label.n
[1] 4
```

```
attr("class")
[1] "L-matrix"
```

Convert back to various formats (not run)

```
> # backto.matching <- toMatching(exam.L.matrix )
> # backto.L.matrix <- toMoment(backto.matching$matching[,c(1,2)])
> # backto.Newick <- toNewick(exam.L.matrix$L,type="square")
> # backto.L.matrix.tips <- toMoment(exam.Newick,tip.label=c("d","a","c","b"))
> # backto.phylo <- as.phylo.matching(backto.matching)
```

References

- L^AT_EX3 Project Team (2009). *L^AT_EX - A Document Preparation System*. The L^AT_EX3 Project. URL <http://www.latex-project.org/>.
- Anderson TW (1971). *An Introduction to Multivariate Statistical Analysis*. John Wiley & Sons, New York.
- Diaconis P, Holmes S (1998). “Matchings and Phylogenetic Trees.” *Proc. Natl. Acad. Sci.*, **95**(25), 14600–14602.
- Felsenstein J (1990). “The Newick tree format.” <http://evolution.genetics.washington.edu/phylip/newicktree.html>.
- Felsenstein J (2004). *Inferring Phylogenies*. Sinauer Associates, Inc., Sunderland.
- Foulds L, Graham R (1982). “The Steiner problem in phylogeny is NP-complete.” *Advances in Applied Mathematics*, **3**, 43–49.
- Genz A, Bretz F, with contributions by Tetsuhisa Miwa TH, Mi X, Leisch F, Scheipl F (2008). *mvtnorm: Multivariate Normal and t Distributions*. R package version 0.9-2.
- Hankin RKS (2010). “multipol: Multivariate Polynomials.” R package version 1.0-4, URL <http://CRAN.R-project.org/package=multipol>.
- Isserlis L (1918). “On a Formula for the Product-Moment Coefficient of Any Order of a Normal Frequency Distribution in Any Number of Variables.” *Biometrika*, **12**(1,2), 134–139.
- Kahle D (2014). “mpoly: Symbolic computation and more with multivariate polynomials.” R package version 0.0.5, URL <http://cran.r-project.org/web/packages/mpoly/mpoly.pdf>.
- Kan R (2008). “From Moments of Sums to Moments of Product.” *Journal of Multivariate Analysis*, **99**, 542–554.
- Muirhead RJ (1982). *Aspects of Multivariate Statistical Theory*. John Wiley & Sons, New York.
- Paradis E, Claude J, Strimmer K (2004). “APE: Analyses of Phylogenetics and Evolution in R Language.” *Bioinformatics*, **20**, 289–290.
- Phillips K (2010). “R Functions to Symbolically Compute the Central Moments of the Multivariate Normal Distribution.” *Journal of Statistical Software, Code Snippets*.

- R Development Core Team (2008). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL <http://www.R-project.org>.
- Rizzo ML (2008). *Statistical Computing with R*. Chapman & Hall/CRC, London.
- Stergiou C, Siganos D (1996). “Neural Networks.” *Technical report*, Department of Computing, Imperial College, Surveys and Presentations in Information Systems Engineering (SURPRISE). URL http://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol4/cs11/report.html.
- Tracy DS, Sultan SA (1993). “Higher Order Moments of Multivariate Normal Distribution Using Matrix Derivatives.” *Stochastic Analysis and Applications*, **11**, 337–348.
- Wikipedia contributors (2009). *Multivariate Normal Distribution*. Date of last revision: 8 December 2009 17:01 UTC, URL http://en.wikipedia.org/w/index.php?title=Multivariate_normal_distribution&oldid=330485779.

Affiliation:

Kem Phillips
PO Box 434
Cavendish
VT 05142 USA
E-mail: kemphillips@comcast.net