

Appendix (A)

The Symmetric Matrix S

Description:

This function calculates the symmetric matrix S .

Usage:

`Smatrix(s, pix)`

Arguments:

`s`: a vector of variables containing the six free parameters in the S matrix.

`pix`: a vector giving the stationary probabilities for the four nucleotides A, C, G and T.

Details:

This function calculates the matrix S , which we used to calculate the rate matrix R .

See Chapter 2 Section 2.1.1 for more details.

Value:

A 4×4 symmetric matrix.

See Also:

`Pt`, `Fmatrix`, `gn`, `gn2`.

Example:

```
s=c(.1,.2,.3,.4,.5,.6)
```

```
pi=c(.1,.1,.1,.7)
```

```
Smatrix(s, pi)
```

	[,1]	[,2]	[,3]	[,4]
[1,]	-2.4	0.1	0.2	0.3
[2,]	0.1	-4.0	0.4	0.5
[3,]	0.2	0.4	-4.8	0.6
[4,]	0.3	0.5	0.6	-0.2

The Transition Probability Function

Description:

This function calculates the transition probability function for a process during a period of time.

Usage:

Pt(S, Pi, t)

Arguments:

S: a 4×4 symmetric matrix.

Pi: a diagonal matrix containing the stationary distribution for the process.

t: a period of time describing the length of the process.

Details:

This function needs the 4×4 symmetric matrix S , Π and the process length t in order to find the transition probability over that process, where $P_{ij}(t)$ is the probability that the i^{th} nucleotide changes to the j^{th} nucleotide during the period of t .

See Chapter 2 Section 2.1 for more details.

Value:

A 4×4 matrix containing the transition probabilities for a process.

See Also:

Smatrix.

Example:

```
Pi=diag(c(.1,.1,.1,.7)), S=Smatrix(c(.3,.3,.3,.3,.3,.3),diag(Pi)), t=1
Pt(S, Pi, t)
```

	[,1]	[,2]	[,3]	[,4]
[1,]	0.76673640	0.02591818	0.02591818	0.1814272
[2,]	0.02591818	0.76673640	0.02591818	0.1814272
[3,]	0.02591818	0.02591818	0.76673640	0.1814272
[4,]	0.02591818	0.02591818	0.02591818	0.9222455

Joint Distribution for Two Matched Sequences

Description:

This function calculates the joint distribution function for two edge tree.

Usage:

Fmatrix(t1, t2, f0, Sx, Sy, Pix, Piy)

Arguments:

t1: represents the length from the tree root to the first node.

t2: represents the length from the tree root to the second node.

f0: the initial distribution for the four nucleotides.

Sx: a 4×4 symmetric matrix related to the first edge.

Sy: a 4×4 symmetric matrix related to the second edge.

Pix: a diagonal matrix for the stationary distribution of the first edge.

Piy: a diagonal matrix for the stationary distribution of the second edge.

Details:

This function calculates the joint distribution function for a two edge tree with different edge lengths, stationary distributions and different S matrices.

See Chapter 2 Example 2.1 for more details.

Value:

A 4×4 matrix containing the joint edges.

See Also:

gn, Smatrix.

Example:

```
f0=c(.25,.25,.25,.25)
```

```
Pi1=diag(c(.2,.2,.2,.4))
```

```
Pi2=diag(c(.1,.1,.1,.7))
```

```
S1=Smatrix(c(.2,.2,.2,.2,.2,.2),diag(Pi1))
```

```
S2=Smatrix(c(.3,.3,.3,.3,.3,.3),diag(Pi2))
```

```
Fmatrix(1, .5, f0, S1, S2, Pi1, Pi2)
```

	[,1]	[,2]	[,3]	[,4]
[1,]	0.18732907	0.01115705	0.01115705	0.03129337
[2,]	0.01115705	0.18732907	0.01115705	0.03129337
[3,]	0.01115705	0.01115705	0.18732907	0.03129337
[4,]	0.01946303	0.01946303	0.01946303	0.21880130

Joint Distribution for K Matched Sequences

Description:

This function calculates the joint distribution array for K matched sequences.

Usage:

`gn(theta, merge2)`

Arguments:

theta: a vector of variables containing the following parameters in this order:

1. the first three parameters from π_X vector,
2. the first three parameters from π_Y vector,
3. the first three parameters from f_0 vector,
4. the six off-diagonal free parameters in the S matrix,
5. a scalar ρ ,
6. a vector of lengths containing $K - 2$ values.

merge2: a $(K - 1) \times 2$ matrix describing the tree topology.

Details:

This function calculates the joint distribution array for a tree with K matched sequences. it uses the following functions: Pt, Fmatrix and Smatrix.

See Chapter 2 Section 2.6.2 for more details.

Value:

A 4^K array containing the joint distribution for the K edges.

See Also:

Fmatrix, Pt, Smatrix.

Example:

```
merge2=matrix(c(-1,-4,-3,2,-2,-5,1,3), 4, 2)
```

```
theta=c(rep(.25,3), rep(.25,3),rep(.25,3), c(.2,.35,.79,.01,.93,.47),  
3,.1,.5,.8)
```

```
gn(theta, merge2)
```

Note: This will give 4^5 array.

Joint Distribution for K Matched Sequences (2)

Description:

This function calculates the joint distribution array for K matched sequences.

Usage:

gn2(theta, merge2)

Arguments:

theta: is a vector of variables containing the following parameters in this order:

1. the first three parameters from π_X vector,
2. the first three parameters from π_Y vector,
3. the first three parameters from f_0 vector,
4. the six off-diagonal free parameters in the S matrix,
5. a $(K - 1) \times 2$ matrix contains the rate at each edge of the K matched sequences.

merge2: a $(K - 1) \times 2$ matrix describing the tree topology.

Details:

This function calculates the joint distribution array for a tree with K matched sequences. it uses the following functions: Pt, Fmatrix and Smatrix.

See Chapter 2 Section 2.6.3 for more details.

Value:

A 4^K array containing the joint distribution for the K edges.

See Also:

Fmatrix, Pt, Smatrix.

Example:

```
merge2=matrix(c(-1,-4,-3,2,-2,-5,1,3), 4, 2)
```

```
rho2=matrix(c(.3,.5,.3,.2,.3,.5,.8,2.7),4,2)
```

```
theta=c(rep(.25,3), rep(.25,3),rep(.25,3), c(.2,.35,.79,.01,.93,.47),rho2)
```

```
gn2(theta, merge2)
```

Note: This will give 4^5 array.

Generating Samples from a Multinomial Distribution

Description:

Generating random DNA samples from a multinomial distribution.

Usage:

Ntml(n, Ft)

Arguments:

n: sample size

Ft: a 4^K array, containing the joint distribution probabilities for K matched sequences.

Details:

This function generates a 4^K DNA array from a multinomial distribution. It depends on the sample size we need to generate and the 4^K joint distribution array of K matched sequences.

See Chapter 3 Section 3.2 for more details.

Value:

A 4^K observed divergence array.

See Also:

simemb, simapp, gn3sim, gn, gn2, Fmatrix.

Example:

```
merge2=matrix(c(-1,-4,-3,2,-2,-5,1,3), 4, 2)
```

```
theta=c(rep(.25,3), rep(.25,3), rep(.25,3), c(.2,.35,.79,.01,.93,.47),  
3,.1,.5,.8)
```

```
F1=gn(theta,merge2)
```

```
Ntml(1000, F1)
```

Note: This will give a 4^5 observed divergence array.

Generating Random DNA Samples using an Embedded Markov Chain

Description:

This function generates random DNA samples using embedded chain.

Usage:

`simemb(theta, seq Length, merge2)`

Arguments:

theta: is a vector of variables containing the following parameters in this order:

1. the first three parameters from π_X vector,
2. the first three parameters from π_Y vector,
3. the first three parameters from f_0 vector,
4. the six off-diagonal free parameters in the S matrix,
5. a scalar ρ ,
6. a vector of lengths containing $K - 2$ values.

merge2: a $(K - 1) \times 2$ matrix describing the tree topology.

seqlength: the length of sequences we need to generate.

Details:

This function generates 4^K DNA array using embedded Markov chain.

It depends on a set of variables theta, the sequence length and a merge matrix describing the tree topology.

See Chapter 3 Section 3.3 for more details.

Value:

A $n \times K$ observed divergence matrix.

See Also:

Ntml, simapp, gn3sim, gn, gn2, Fmatrix.

Example:

```
theta=(c(rep(.25,3), rep(.25,3), rep(.25,3), c(.2,.35,.79,.01,.93,.47),  
3,.1,.5,.8))
```

```
n=1000
```

```
merge2=matrix(c(-1,-4,-3,2,-2,-5,1,3), 4, 2)
```

```
simemb(theta, n, merge2)
```

Note: This will give 4^5 observed divergence array.

Generating Random DNA Samples using the Rambaut and Grassly Method

Description:

This function generates random DNA samples using Rambaut and Grassly method.

Usage:

```
gn3sim(theta, seq Length, merge2)
```

Arguments:

theta: a vector of variables containing the following parameters in this order:

1. the first three parameters from π_X vector,
2. the first three parameters from π_Y vector,
3. the first three parameters from f_0 vector,
4. the six off-diagonal free parameters in the S matrix,
5. a scalar ρ ,
6. a vector of lengths containing $K - 2$ values.

merge2: a $(K - 1) \times 2$ matrix describing the tree topology.

seqlength: the length of sequences we need to generate.

Details:

This function generates a 4^K DNA array using Rambaut and Grassly, (1997) method. It depends on a set of variables theta, the sequence length and a merge matrix describing the tree topology.

See Chapter 3 Section 3.3.1 for more details.

divergence array.

Value:

A $n \times K$ observed divergence matrix.

See Also:

Ntml, simapp, simemb, gn, gn2, Fmatrix.

Example:

```
theta=(c(rep(.25,3), rep(.25,3), rep(.25,3), c(.2,.35,.79,.01,.93,.47),  
3,.1,.5,.8))
```

```
n=1000
```

```
merge2=matrix(c(-1,-4,-3,2,-2,-5,1,3), 4, 2)
```

```
gn3sim(theta, n, merge2)
```

Note: This will give 4^5 observed divergence array.

Generating Random DNA Samples using an Approximation Method

Description:

This function generates random DNA samples using an approximation method.

Usage:

```
simapp(theta, seq Length, merge2)
```

Arguments:

theta: a vector of variables containing the following parameters in this order:

1. the first three parameters from π_X vector,
2. the first three parameters from π_Y vector,
3. the first three parameters from f_0 vector,
4. the six off-diagonal free parameters in the S matrix,
5. a scalar ρ ,
6. a vector of lengths containing $K - 2$ values.

merge2: a $(K - 1) \times 2$ matrix describing the tree topology.

seqlength: the length of sequences we need to generate.

Details:

This function generates a 4^K DNA array using an approximation method.

It depends on a set of variables theta, the sequence length and a merge matrix describing the tree topology.

See Chapter 3 Section 3.4 for more details.

Value:

A $n \times K$ observed divergence matrix.

See Also:

Ntml, simemb, gn3sim, gn, gn2, Fmatrix.

Example:

```
theta=(c(rep(.25,3), rep(.25,3), rep(.25,3), c(.2,.35,.79,.01,.93,.47),  
3,.1,.5,.8))
```

```
n=1000
```

```
merge2=matrix(c(-1,-4,-3,2,-2,-5,1,3), 4, 2)
```

```
simapp(theta, n, merge2)
```

Note: This will give 4^5 observed divergence array.

Transforming 4^K Array to $m \times K$ Matrix

Description:

This function transfers any array to a matrix.

Usage:

artomat(Ft)

Arguments:

Ft: a 4^K array, containing the observed divergent frequencies for K matched sequences.

Details:

This function transfers any 4^K array containing the observed divergent frequencies of K matched sequences to an $m \times K$ matrix, where m is the sum of the frequencies in the 4^K observed divergence array.

See Chapter 3, Example 3.1 for more details.

Value:

An $m \times K$ matrix, where m is the sum of the frequencies in the 4^K divergence

See Also:

gn2, gn, Fmatrix.

Example:

```
F1=gn(c(rep(.25,3), rep(.25,3), rep(.25,3), c(.2,.35,.79,.01,.93,.47),
3,.1,.5,.8))
```

```
N1=Ntml(1000,F1)
```

```
artomat(N1)
```

Note: This will give 1000×5 matrix.

Paralinear Distances

Description:

This function calculates the paralinear distance between K matched DNA sequences.

Usage:

Distance(F4)

Arguments:

F4: a 4^K array containing the joint distribution array $F(t)$ or the observed array N .

Details:

This function calculates the paralinear distances between K matched DNA sequences, depending on the joint distribution array for these K sequences or on the observed divergence array N .

See Chapter 5 Section 5.2 for more details.

Value:

A $K \times K$ symmetric matrix distances between the K sequences.

See Also:

gn2, gn, Fmatrix, Ntml.

Example:

```
merge2=matrix(c(-1,-4,-3,2,-2,-5,1,3), 4, 2)
theta=c(rep(.25,3), rep(.25,3), rep(.25,3), c(.2,.35,.79,.01,.93,.47), 3,.1,.5,.8)
F1=gn(theta,merge2)
Distance(F1)
```

	[,1]	[,2]	[,3]	[,4]	[,5]
[1,]	0.000	0.825	6.6	5.500	5.500
[2,]	0.825	0.000	6.6	5.500	5.500
[3,]	6.600	6.600	0.0	5.500	5.500
[4,]	5.500	5.500	5.5	0.000	1.375
[5,]	5.500	5.500	5.5	1.375	0.000

Test for Symmetry of Matched DNA Sequences

Description:

This function tests for symmetry between all the pairs of K matched DNA sequences.

Usage:

TEST2(Nt)

Arguments:

Nt: a 4^K array containing the observed divergence array N .

Details:

This function calculates Bowker's test for symmetry, Stuart's test for marginal symmetry and the test for internal symmetry. It depends on the 4^K observed divergence array N .

See Chapter 4 Sections 4.2, 4.3 and 4.4, for more details.

Value:

A list of three lower triangle matrices:

1. the lower triangle of the matrix contains $(K - 1) \times (K - 1)$ values shows Bowker's test between all the possible pairs of the K sequences.
2. the lower triangle of the matrix contains $(K - 1) \times (K - 1)$ values shows Stuart's test between all the possible pairs of the K sequences.
3. the lower triangle of the matrix contains $(K - 1) \times (K - 1)$ values shows the internal test between all the possible pairs of the K sequences.

See Also:

Ntml, simapp, simemb, TEST3.

Example:

```
merge2=matrix(c(-1,-4,-3,2,-2,-5,1,3), 4, 2)
theta=c(rep(.25,3), rep(.25,3), rep(.25,3), c(.2,.35,.79,.01,.93,.47),
3,.1,.5,.8)
F1=gn(theta,merge2)
N1=Ntml(1000,F1)
TEST2(N1)

$Bowker.test
      1      2      3      4
2 0.5757
3 0.1403 0.7638
4 0.3291 0.7729 0.7252
5 0.8371 0.9933 0.8939 0.4990

$Stuart.test
      1      2      3      4
2 0.4860
3 0.3978 0.7549
4 0.2637 0.4012 0.7752
5 0.7695 0.9763 0.9110 0.2742

$Internal.test
      1      2      3      4
2 0.5101
3 0.0825 0.5402
4 0.4022 0.9520 0.4694
5 0.6507 0.9088 0.6306 0.6888
```

Overall Test for Marginal Symmetry

Description:

This function tests for symmetry between K matched DNA sequences.

Usage:

```
TEST3(Nt)
```

Arguments:

Nt: a 4^K array containing the observed divergence array N .

Details:

This function calculates overall test for marginal symmetry. It depends on the 4^K observed divergence array N .

See Chapter 4 Sections 4.5, for more details.

Value:

A single value gives the overall test for marginal symmetry between K matched sequences.

See Also:

Ntml, simapp, simemb, TEST2.

Example:

```
merge2=matrix(c(-1,-4,-3,2,-2,-5,1,3), 4, 2)
theta=c(rep(.25,3), rep(.25,3), rep(.25,3), c(.2,.35,.79,.01,.93,.47),
3,.1,.5,.8)
F1=gn(theta,merge2)
N1=Ntml(1000,F1)
TEST3(N1)
[1] 9.217612
```

Negative Log Likelihood Ratio

Description:

This function calculates log likelihood ratio value.

Usage:

```
likelihood(thetast,fobs,merge2)
```

Arguments:

thetast: a starting values for the parameter we need to estimate.

fobs: the 4^K joint distribution array for K edge tree.

merge2: a $(K - 1) \times 2$ matrix describing the tree topology.

Details:

This function calculates the log likelihood ratio value for $F(t)$. It needs a vector of starting values for the parameters estimate, 4^K observed divergence array and merge matrix describing the tree topology.

See Chapter 6 Sections 6.2, for more details.

Value:

The value of the log likelihood ratio.

See Also:

gn, gn2.

Example:

```
merge2=matrix(c(-1,-4,-3,2,-2,-5,1,3), 4, 2)
theta=c(rep(.25,3), rep(.25,3), rep(.25,3), c(.2,.35,.79,.01,.93,.47),
3,.1,.5,.8)
F1=gn(theta,merge2)
likelihood(theta,F1,merge2)
[1] 0.00
```