Package ‘DNAsseqtest’
March 24, 2016

Type Package
Title Generating and Testing DNA Sequences
Version 1.0
Date 2016-03-26
Author Faisal Ababneh, John Robinson, Lars S Jermiin and Hasinur Rahaman Khan
Maintainer Hasinur Rahaman Khan <hasinurkhan@gmail.com>
Description Generates DNA sequences based on Markov model techniques for matched sequences. This can be generalized to several sequences. The sequences (taxa) are then arranged in an evolutionary tree (phylogenetic tree) depicting how taxa diverge from their common ancestors. This gives the tests and estimation methods for the parameters of different models. Standard phylogenetic methods assume stationarity, homogeneity and reversibility for the Markov processes, and often impose further restrictions on the parameters.
License GPL-2
NeedsCompilation no
Repository CRAN
Date/Publication 2016-03-24 23:32:09

R topics documented:

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>DNAsseqtest-package</td>
<td>2</td>
</tr>
<tr>
<td>artomat</td>
<td>3</td>
</tr>
<tr>
<td>Distance</td>
<td>4</td>
</tr>
<tr>
<td>Fmatrix</td>
<td>5</td>
</tr>
<tr>
<td>gn</td>
<td>6</td>
</tr>
<tr>
<td>gn2</td>
<td>7</td>
</tr>
<tr>
<td>gn3sim</td>
<td>8</td>
</tr>
<tr>
<td>likelihood</td>
<td>9</td>
</tr>
<tr>
<td>Ntml</td>
<td>10</td>
</tr>
<tr>
<td>Pt</td>
<td>11</td>
</tr>
<tr>
<td>simapp</td>
<td>12</td>
</tr>
<tr>
<td>simemb</td>
<td>13</td>
</tr>
<tr>
<td>Smatrix</td>
<td>14</td>
</tr>
<tr>
<td>TEST2</td>
<td>15</td>
</tr>
<tr>
<td>TEST3</td>
<td>16</td>
</tr>
</tbody>
</table>
DNAseqtest-package

Generating and Testing DNA Sequences

Description
Generates DNA sequences based on Markov model techniques for matched sequences. This can be generalized to several sequences. The sequences (taxa) are then arranged in an evolutionary tree (phylogenetic tree) depicting how taxa diverge from their common ancestors. This gives the tests and estimation methods for the parameters of different models. Standard phylogenetic methods assume stationarity, homogeneity and reversibility for the Markov processes, and often impose further restrictions on the parameters.

Details

Package: DNAseqtest
Type: Package
Version: 1.0
Date: 2016-03-26
License: GPL-2

Author(s)
Faisal Ababneh, John Robinson, Lars S Jermiin and Hasinur Rahaman Khan
Maintainer: Hasinur Rahaman Khan <hasinurkhan@gmail.com>

References

Examples
# To generate a 4*5 gene array
merge2<-matrix(c(-1L-TL-SL2L-2L-5L1LS)LTL2)
theta<-c(rep(N25LS)L rep(N25LS)L rep(N25LS)L c(N2LNS5LN79LNP1LN9SLNT7)L SLN1LN5LN8)
ecn<-gn(thetaL merge2)
Description

This function transfers any array to a matrix.

Usage

artomat(fobs)

Arguments

fobs 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.

Value

An \(m \times K\) matrix

References


See Also

gn2, gn, Fmatrix
### Distance

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

### Usage

```r
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 $N$.

### Value

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

### References


### See Also

- `gn2`, `gn`, `Fmatrix`, `Ntml`

### Examples

```r
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)
dn<-Distance(F1)
dn
```
**Fmatrix**

*Joint Distribution for Two Matched Sequences*

**Description**

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

**Usage**

```
Fmatrix(t1, t2, f0, Sx2, Sy2, 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
- `Sx2`: a 4 x 4 symmetric matrix related to the first edge
- `Sy2`: a 4 x 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.

**Value**

A 4 x 4 matrix containing the joint edges

**References**


**See Also**

`gn`, `Smatrix`
Examples

```r
f0<-c(0.25,0.25,0.25,0.25)
Pi1<-diag(c(2,2,2,4))
Pi2<-diag(c(1,1,1,7))
S1<-Smatrix(c(2,2,2,2,2,2,2,2),diag(Pi1))
S2<-Smatrix(c(3,3,3,3,3,3,3,3),diag(Pi2))
fm<-Fmatrix(1, 0.5, S0, S1, S2, Pi1, Pi2)
fm
```

---

**Joint Distribution for K Matched Sequences**

**Description**

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

**Usage**

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

**Arguments**

- `theta`: a vector of variables containing the following parameters in this order: 1. the first three parameters from \( \pi_X \) vector, 2. the first three parameters from \( \pi_Y \) vector, 3. the first three parameters from \( f_0 \) vector, 4. the six off-diagonal free parameters in the \( S \) matrix, 5. a scalar \( \rho \), 6. a vector of lengths containing K-2 values
- `merge2`: (K-1) x 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.

**Value**

A \( 4^K \) array containing the joint distribution for the K edges

**References**


See Also
Fmatrix, Pt, Smatrix

Examples

# To generate a 4^5 gene array
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,.3,.79,.01,.93,.47), 3,.1,.5,.8)
orn.sec<-gn(theta, merge2)
orn.sec

Description
This function calculates the joint distribution array for K matched sequences (second option).

Usage
 gn2(theta, merge2)

Arguments
 theta a vector of variables containing the following parameters in this order–1. the first three parameters from $\pi_X$ vector, 2. the first three parameters from $\pi_Y$ vector, 3. the first three parameters from $f_0$ vector, 4. the six off-diagonal free parameters in the S matrix, 5. a scalar $\rho$, 6. a vector of lengths containing K-2 values
 merge2 (K-1) x 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.

Value
A $4^K$ array containing the joint distribution for the K edges

References
See Also
Fmatrix, Pt, Smatrix

Examples
#To generate a 4*5 gene array
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(1,25,3, rep(.25,3), rep(.25,3), rep(.25,3), c(.2,.35,.79,.01,.93,.47),rho2)
gn2<gn2(theta, merge2)
gn2

gn3sim 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, seqLength, merge2)

Arguments
theta a vector of variables containing the following parameters in this order:
  1. the first three parameters from $\pi_X$ vector, 2. the first three parameters from $\pi_Y$ vector,
  3. the first three parameters from $f_0$ vector, 4. the six off-diagonal free parameters in the S matrix,
  5. a scalar $\rho$, 6. a vector of lengths containing K-2 values
seqLength the length of sequences we need to generate
merge2 (K-1) x 2 matrix describing the tree topology

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.

Value
A n x K observed divergence matrix

References
**likelihood**  

*Negative Log Likelihood Ratio*

**Description**

This function calculates log likelihood ratio value.

**Usage**

```r
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) x 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.

**Value**

The value of the log likelihood ratio

**References**

See Also

gn, gn2

Examples

```r
c(-1L, -4L, -3L, 2L, -2L, -5L, 1L, 3L), 4L, 2L)
theta <- c(rep(.25L, 3L), rep(.25L, 3L), c(.2L, .35L, .79L, .01L, .93L, .47L), 3L, 1L, 5L, 8L)
F1 <- gn(theta, merge2)
ln <- likelihood(theta, F1, merge2)
ln
```
Examples

# This will give a 4*K observed divergence array
merge2<-matrix(c(-1L-TL-SL2L-2L-5L1LS)L TL 2)
theta<-c(rep(N25LS)L rep(N25LS)L rep(N25LS)L c(N2LNS5LN79LNP1LN9SLNT7)L
SLN1LN5LN8)
f1<-gn(thetaLmerge2)
Nt<-Ntml(1000, F1)
Nt

---

Pt          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 x 4 symmetric matrix
Pi    a diagonal matrix containing the stationary distribution for the process

Details

This function needs the 4 x 4 symmetric matrix S, II 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 $t$.

Value

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

References


See Also

Smatrix
Examples

```r
pi <- diag(c(1, 1, 1, 1, 1))
S <- smatrix(c(3, 3, 3, 3, 3, 3), diag(pi))
t <- 1
p <- pt(S, pi, t)
p
```

**Description**

This function generates random DNA samples using an approximation method.

**Usage**

```r
simapp(theta, seqLength, merge1)
```

**Arguments**

- `theta` a vector of variables containing the following parameters in this order: 1. the first three parameters from $\pi_X$ vector, 2. the first three parameters from $\pi_Y$ vector, 3. the first three parameters from $f_0$ vector, 4. the six off-diagonal free parameters in the S matrix, 5. a scalar $\rho$, 6. a vector of lengths containing $K-2$ values.
- `seqLength` the length of sequences we need to generate
- `merge1` $(K-1) \times 2$ matrix describing the tree topology

**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.

**Value**

A $n \times K$ observed divergence matrix

**References**


**See Also**

Ntml, simemb, gn3sim, gn, gn2, Fmatrix
Examples

```r
# This will give 4^5 observed divergence array
theta<-c(rep(.25,3), rep(.25,3), rep(.25,3), c(.2,.2,.2,.2,.2,.2),
3,.1,.5,.8))
n<-1000
merge2<-matrix(c(-1,-4,-3,2,-2,-5,1,3), 4, 2)
sa<-simapp(theta, n, merge2)
sa
```

Description

This function generates random DNA samples using embedded Markov chain.

Usage

```r
simemb(theta, seqlength, merge2)
```

Arguments

- **theta**: a vector of variables containing the following parameters in this order: 1. the first three parameters from \( \pi_X \) vector, 2. the first three parameters from \( \pi_Y \) vector, 3. the first three parameters from \( f_0 \) vector, 4. the six off-diagonal free parameters in the S matrix, 5. a scalar \( \rho \), 6. a vector of lengths containing K-2 values
- **seqlength**: the length of sequences we need to generate
- **merge2**: (K-1) x 2 matrix describing the tree topology

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.

Value

A n x K observed divergence matrix

References


See Also

Ntml, simapp, gn3sim, gn, gn2, Fmatrix
Examples

# This will give 4*5 observed divergence array
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)
sm<-simemb(theta, n, merge2)

---------

Smatrix  
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.

Value

A 4 x 4 symmetric matrix

References


See Also

Pt, Fmatrix, gn, gn2

Examples

s<-c(.1,.2,.3,.4,.5,.6)
pi<-c(.1,.1,.1,.7)
sm<-Smatrix(s, pi)
sm
**Test for Symmetry of Matched DNA Sequences**

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

**Usage**
```r
TEST2(f)
```

**Arguments**
- `f` 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 $N$.

**Value**
A list of three lower triangle matrices
- **first** 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
- **second** 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
- **third** 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

**References**

**See Also**
- Ntml, simapp, simemb, TEST3
Examples

```r
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 <- Ntm1(1000, F1)
t2 <- TEST2(N1)
t2
```

---

TEST3

*Overall Test for Marginal Symmetry*

**Description**

This function tests for symmetry between K matched DNA sequences.

**Usage**

```r
TEST3(Farray)
```

**Arguments**

- `Farray` 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 $N$.

**Value**

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

**References**


**See Also**

Ntm1, simapp, simemb, TEST2
Examples

```r
merge2 <- matrix(c(-1, -4, -3, 2, -2, -5, 1, 3), 4, 2)
theta <- c(rep(2, 3), rep(2, 3), rep(2, 3), c(2, .35, .79, .01, .93, .47),
3, 1, 5, 8)
F1 <- gn(theta, merge2)
N1 <- ntm1(1000, F1)
t3 <- TEST3(N1)
t3
```
## Index

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Array</td>
<td>artomat. 3</td>
</tr>
<tr>
<td>Distance between sequences</td>
<td>Distance. 4</td>
</tr>
<tr>
<td>Generate sequence</td>
<td>gn3sim. 8</td>
</tr>
<tr>
<td></td>
<td>simapp. 12</td>
</tr>
<tr>
<td></td>
<td>simemb. 13</td>
</tr>
<tr>
<td>Generated sample</td>
<td>Ntml. 10</td>
</tr>
<tr>
<td>Joint distribution</td>
<td>Fmatrix, 5</td>
</tr>
<tr>
<td></td>
<td>gn. 6</td>
</tr>
<tr>
<td></td>
<td>gn2. 7</td>
</tr>
<tr>
<td>Likelihood</td>
<td>likelihood. 9</td>
</tr>
<tr>
<td>Sequence test</td>
<td>DNAseqtest-package, 2</td>
</tr>
<tr>
<td></td>
<td>TEST2, 15</td>
</tr>
<tr>
<td></td>
<td>TEST3, 16</td>
</tr>
<tr>
<td>Symmetry matrix</td>
<td>Smatrix, 14</td>
</tr>
<tr>
<td>Transition probability</td>
<td>Pt, 11</td>
</tr>
</tbody>
</table>

artomat, 3

Distance, 4
DNAseqtest (DNAseqtest-package), 2
DNAseqtest-package, 2

Fmatrix, 5

gn, 6
gn2, 7
gn3sim, 8

likelihood, 9

Ntml, 10