

# Package: phylometrics (via r-universe)

August 28, 2024

**Title** Estimating Statistical Errors of Phylogenetic Metrics

**Version** 0.0.1

**Author** Xia Hua <xia.hua@anu.edu.au>, Lindell Bromham  
<lindell.bromham@anu.edu.au>

**Maintainer** Xia Hua <xia.hua@anu.edu.au>

**Description** Provides functions to estimate statistical errors of phylogenetic metrics particularly to detect binary trait influence on diversification, as well as a function to simulate trees with fixed number of sampled taxa and trait prevalence.

**Depends** R (>= 2.1.1)

**Imports** mvtnorm

**License** GPL (>= 2)

**Suggests** testthat

**RoxygenNote** 5.0.1

**NeedsCompilation** no

**Date/Publication** 2015-11-17 14:40:44

**Repository** <https://amsibioinfo.r-universe.dev>

**RemoteUrl** <https://github.com/cran/phylometrics>

**RemoteRef** HEAD

**RemoteSha** 5029de608db4a1bcdd8823da24ddf2b111e75ee1

## Contents

fpd . . . . .	2
noto . . . . .	2
sscd . . . . .	3
tars . . . . .	3
treesim . . . . .	4
treestat . . . . .	4

<b>Index</b>	<b>6</b>
--------------	----------

---

fpd	<i>calculate the Fritz &amp; Purvis D (FPD) metric</i>
-----	--

---

**Description**

This function calculates the FPD metric. To conduct significance test on FPD, use 'fpd' as input of 'func' in 'treestat'

**Usage**

```
fpd(state, phy)
```

**Arguments**

state	a vector of '0' and '1' for trait state of each tip, in the same order as the tip labels.
phy	an object of class 'phylo'.

---

noto	<i>calculate the number of tips per origin (NoTO) metric</i>
------	--

---

**Description**

This function calculates the NoTO metric. To conduct significance test on NoTO, use 'noto' as input of 'func' in 'treestat'

**Usage**

```
noto(state, phy)
```

**Arguments**

state	a vector of '0' and '1' for trait state of each tip, in the same order as the tip labels.
phy	an object of class 'phylo'.

---

sscd	<i>calculate the sum of sister clade differences (SSCD) metric</i>
------	--

---

**Description**

This function calculates the SSCD metric. To conduct significance test on SSCD, use 'sscd' as input of 'func' in 'treestat'

**Usage**

```
sscd(state, phy)
```

**Arguments**

state	a vector of '0' and '1' for trait state of each tip, in the same order as the tip labels.
phy	an object of class 'phylo'.

---

tars	<i>calculate the tip age rank sum (TARS) metric</i>
------	---

---

**Description**

This function calculates the TARS metric. To conduct significance test on TARS, use 'tars' as input of 'func' in 'treestat'

**Usage**

```
tars(state, phy)
```

**Arguments**

state	a vector of '0' and '1' for trait state of each tip, in the same order as the tip labels.
phy	an object of class 'phylo'.

---

treesim *Simulate trees with fixed number of sampled taxa and trait prevalence*

---

### Description

This function generates a tree that contains a defined number of sampled taxa for each of the two trait states.

### Usage

```
treesim(pars, N0, N1, sampling.f, max.t = Inf)
```

### Arguments

pars	a vector of parameters that describe the macroevolutionary processes of the tree. Parameters are in order of speciation rate for trait state 0, speciation rate for trait state 1, extinction rate for trait state 0, extinction rate for trait state 1, transition rate from state 0 to state 1, transition rate from state 1 to state 0.
N0	the number of sampled taxa with trait state 0.
N1	the number of sampled taxa with trait state 1.
sampling.f	a vector of sampling fraction of taxa with trait state 0 and trait state 1.
max.t	the maximum amount of time, above which tree simulation stops and reports the tree as not being able to coalesce.

### Examples

```
phy <- treesim(pars=c(0.1,0.1,0.05,0.05,0.1,0.1),N0=50,N1=50,sampling.f=c(1,1),max.t=Inf)
```

---

treestat *Significance test of user defined phylogenetic metric*

---

### Description

This function conducts the significance test of a phylogenetic metric defined by users.

### Usage

```
treestat(phy, stlist = NULL, state = NULL, func, par = NULL,
         traitevol = NULL, a = NULL, alternative = "two.sided", simplify = T)
```

**Arguments**

phy	an object of class 'phylo'.
stlist	a vector of tip labels with trait state '1'.
state	a vector of '0' and '1' for trait state of each tip, in the same order as the tip labels.
func	name of the function that calculates the metric. The first two inputs of the function need to be the state vector and the phylo class.
par	values of additional parameters used in the above function.
traitevol	the null model of trait evolution used in the significance test of the metric. <code>traitevol = "TBM"</code> is the threshold brownian motion with the same trait prevalence as observed. <code>traitevol = "random"</code> is the random reshuffle of tip states. If users only want to calculate the value of the metric or if the null model is defined in the 'func' already, <code>traitevol = NULL</code> .
a	number of traits simulated under the null model of trait evolution.
alternative	the alternative hypothesis for the significant test of the metric. <code>alternative = "greater"</code> , if users want to test if the observed trait has significantly larger metric value than expected by the null model of trait evolution. <code>alternative = "less"</code> , if users want to test if the observed trait has significantly smaller metric value. <code>alternative = "two.sided"</code> (default), if users have no prior knowledge on how the observed trait may differ from the null model.
simplify	if TRUE, the output is simplified. The full output includes the p.value, metric value of the observed trait, metric values of the simulated traits, a matrix, each row of which is the state vector of a simulated trait. The simplified output includes the p.value.

**Details**

In the example, we conduct significance test on a metric defined by 'tars' function for a simulated phylogeny. The null hypothesis is that the trait is randomly distributed across tips. We don't know if the trait will have larger or smaller metric value than expected, so we apply a "two.sided" test. 1000 traits are simulated to generate the null distribution of the metric. The state vector is included in the phylo class as the output of treesim, so we don't need to input state here.

**Examples**

```
phy <- treesim(pars=c(0.1,0.1,0.05,0.05,0.1,0.1),N0=50,N1=50,sampling.f=c(1,1),max.t=Inf)
treestat(phy, func=tars, traitevol="random", a=1000, alternative="two.sided", simplify=TRUE)
```

# Index

fpd, 2

noto, 2

sscd, 3

tars, 3

treesim, 4

treestat, 4