

# Package ‘pcrcoal’

September 9, 2016

**Version** 1.2.0

**Date** 2016-09-04

**Title** Implementing the Coalescent Approach to PCR Simulation Developed  
by Weiss and Von Haeseler (NAR, 1997)

**Author** Botond Sipos <[sbotond@gmail.com](mailto:sbotond@gmail.com)>

**Maintainer** Botond Sipos <[sbotond@gmail.com](mailto:sbotond@gmail.com)>

**Depends** R (>= 2.11.1), ape (>= 2.3), ggplot2 (>= 0.8.8), methods (>= 2.11.1), phylosim (>= 0.12)

**Suggests**

**Imports**

**Description** Implementing the Coalescent Approach to PCR Simulation.

**License** GPL (>= 3)

**URL** <http://github.com/bsipos/pcrcoal>

**LazyLoad** TRUE

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2016-09-09 00:41:18

## R topics documented:

PCRcoal-class . . . . .	2
sample.tnt . . . . .	4
sample.tree . . . . .	6
sample.trs . . . . .	7

<b>Index</b>	<b>9</b>
--------------	----------

---

PCRcoal-class

*The PCRcoal class*

---

### Description

PCRcoal is a class implementing the coalescent approach to PCR simulation developed by Weiss and von Haeseler (NAR, 1997).

### Usage

```
PCRcoal( initial.size,nr.cycles,efficiencies,sample.size,max.tries=100 )
```

Returns a PCRcoal S4 object.

### Slots

`initial.size`: Number of template molecules.

`nr.cycles`: Number of PCR cycles.

`efficiencies`: A vector of per-cycle PCR efficiencies. It is recycled if its length is smaller than `nr.cycles`.

`sample.size`: Number of molecules sampled after amplification.

`max.tries`: Maximum number of tries when sampling a size trajectory with a large enough final size.

### Methods

`sample.tree`: Sample a tree, return as an [ape](#) phylo object.

`sample.tnt`: Return the sampled tree, subsample sizes and size trajectories in a list.

`sample.trs`: Return the sampled subsample sizes and size trajectories in a list.

### Note

- The original method of Weiss and von Haeseler produces a tree containing a multifurcating root node connected to the subtrees corresponding to the different subsamples. In order to be able to store the output in phylo objects, PCRcoal randomly breaks the polytomy by "coalescing" the subsample trees with zero branch lengths. Some branch length might be non-zero, in the case a subsample coalesced into one molecule which accumulated branch length ("replication count") in the previous cycles.
- If the cycle number and/or the efficiencies are small, it can happen that the number of the molecules after the amplification is less than the specified sample size. PCRcoal will try sampling size trajectories `max.tries` times (100 by default) to get enough amplified molecules and then aborts the simulation.

### Author(s)

Botond Sipos

## References

Weiss, G, von Haeseler, A (1997) A coalescent approach to the polymerase chain reaction. NAR 25(15): 3082-3087 <http://dx.doi.org/10.1093/nar/25.15.3082>

## See Also

The [ape](#) and [phylosim](#) packages.

## Examples

```
# Construct a PCRcoal object
# and specify the experimental conditions:
sim <-PCRcoal(
  initial.size   =5,
  sample.size   =10,
  nr.cycles     =20,
  efficiencies   =c(rep(0.1, 30))
)

##
## Method: sample.tree
##

# Sample a coalescent tree:
tree <- sample.tree(sim)

# Print tree info:
print(tree)

# Plot the tree:
plot(tree)

##
## Method: sample.tnt
##

# Sample a coalescent tree with subsample sizes and
# size trajectories:
res <- sample.tnt(sim)

# Print subsample sizes:
print(res$subsamples)

# Print size trajectories:
print(res$trajectories)

# Print tree info:
print(res$phylo)

##
## Method: sample.trs
##
```

```

# Sample subsample sizes and size trajectories:
res <- sample.trs(sim)

# Print subsample sizes:
print(res$subsamples)

# Print size trajectories:
print(res$trajectories)

##
## Simulating mutations using PhyloSim
##

# Sample tree:
tree <- sample.tree(sim)

# Load the phylosim package
library(phylosim)
# Construct a PhyloSim object, set up the root
# sequence and the substitution process:
psim <- PhyloSim(
  phylo = tree,
  root.seq=sampleStates( NucleotideSequence(length=10,processes=list(list(JC69())) ) )
)
# Scale the tree:
scaleTree(psim,0.1)

# Simulate mutations:
Simulate(psim)

# Plot the resulting alignment:
plot(psim,plot.ancestors=FALSE)

```

---

sample.tnt

*Sample a coalescent tree according to a PCR experiment, return also subsample sizes and size trajectories*

---

### Description

This method returns a coalescent tree sampled according to the PCR experiment specified by the parameters stored in the slots of the [PCRcoal](#) object given as the first argument. All of the slots must contain a defined value.

This method also returns the subsample sizes and size trajectories corresponding to the sampled tree.

### Usage

```

## S4 method for signature 'PCRcoal'
sample.tnt(object)

```

**Arguments**

object            A [PCRcoal](#) object, with all the slots defined.

**Value**

A list containing the following elements:

**phylo** A [ape](#) phylo object (rooted).

**trajectories** A matrix containing the sampled size trajectories (rows ~ subsamples, columns ~ cycles).

**subsamples** A vector of subsample sizes.

**See Also**

The [PCRcoal](#) class. The [sample.tree](#) and [sample.trs](#) methods.

**Examples**

```
# Construct a PCRcoal object
# and specify the experimental conditions:
sim <-PCRcoal(
  initial.size   =5,
  sample.size    =10,
  nr.cycles      =30,
  efficiencies    =c(rep(0.1, 30))
)

# Sample a coalescent tree with subsample sizes and
# size trajectories:
res <- sample.tnt(sim)

# Print subsample sizes:
print(res$subsamples)

# Print size trajectories:
print(res$trajectories)

# Print tree info:
print(res$phylo)

# Plot the tree:
plot(res$phylo)
```

---

`sample.tree`*Sample a coalescent tree according to a PCR experiment*

---

### Description

This method returns a coalescent tree sampled according to the PCR experiment specified by the parameters stored in the slots of the `PCRcoal` object given as the first argument. All of the slots must contain a defined value.

Use this method if you are interested only in the sampled tree.

### Usage

```
## S4 method for signature 'PCRcoal'  
sample.tree(object)
```

### Arguments

`object`            A `PCRcoal` object, with all the slots defined.

### Value

An `ape` phylo object (rooted).

### See Also

The `PCRcoal` class. The `sample.tnt` and `sample.trs` methods.

### Examples

```
# Construct a PCRcoal object  
# and specify the experimental conditions:  
sim <- PCRcoal(  
  initial.size = 5,  
  sample.size = 10,  
  nr.cycles = 30,  
  efficiencies = c(rep(0.1, 30))  
)  
  
# Sample a coalescent tree:  
tree <- sample.tree(sim)  
  
# Print tree info:  
print(tree)  
  
# Plot the tree:  
plot(tree)
```

---

sample.trrs	<i>Sample subsample sizes and size trajectories according to a PCR experiment</i>
-------------	---

---

## Description

This method also returns the subsample sizes and size trajectories sampled according to the PCR experiment specified by the parameters stored in the slots of the `PCRcoal` object given as the first argument.

This method does not sample the tree, which makes it faster than the `sample.tree` method. Use this method if you are interested only in sampling subsamples and size trajectories.

## Usage

```
## S4 method for signature 'PCRcoal'
sample.trrs(object)
```

## Arguments

`object` A `PCRcoal` object, with all the slots defined.

## Value

A list containing the following elements:

**trajectories** A matrix containing the sampled size trajectories (rows ~ subsamples, columns ~ cycles).

**subsamples** A vector of subsample sizes.

## See Also

The `PCRcoal` class. The `sample.tree` and `sample.tnt` methods.

## Examples

```
# Construct a PCRcoal object
# and specify the experimental conditions:
sim <- PCRcoal(
  initial.size = 5,
  sample.size = 10,
  nr.cycles = 30,
  efficiencies = c(rep(0.1, 30))
)

# Sample subsample sizes and size trajectories:
res <- sample.trrs(sim)

# Print subsample sizes:
```

```
print(res$subsamples)

# Print size trajectories:
print(res$trajectories)
```



# Index

\*Topic **PCR, simulation**

sample.tnt, [4](#)  
sample.tree, [6](#)  
sample.trs, [7](#)

\*Topic **classes, coalescent, PCR, simulation**

PCRcoal-class, [2](#)

ape, [2](#), [3](#), [5](#), [6](#)

PCRcoal, [4–7](#)

PCRcoal (PCRcoal-class), [2](#)

PCRcoal,missing-method (PCRcoal-class),  
[2](#)

PCRcoal-class, [2](#)

phylosim, [3](#)

sample.tnt, [4](#), [6](#), [7](#)

sample.tnt,PCRcoal-method (sample.tnt),  
[4](#)

sample.tree, [5](#), [6](#), [7](#)

sample.tree,PCRcoal-method  
(sample.tree), [6](#)

sample.trs, [5](#), [6](#), [7](#)

sample.trs,PCRcoal-method (sample.trs),  
[7](#)