

Dinâmica neutra baseada na Teoria Neutra da Biodiversidade, de Hubbell.

Description :

`hubbells.game()` realiza simulações de uma dada comunidade (cujas condições iniciais são determinadas pelo usuário), que é submetida a uma dinâmica neutra baseada somente em eventos estocásticos de morte e nascimento de indivíduos que compõem um jogo de soma zero. A função retorna um gráfico com a distribuição de abundância final da comunidade e outro com a variação da riqueza ao longo dos ciclos, além da média do número de espécies final de todas as simulações.

Usage :

```
hubbells.game(s, a, nc, dc, nsim, details=FALSE)
```

Arguments :

<code>s</code>	Numérico. Número inicial de espécies (riqueza) existente na comunidade. Deve ser maior ou igual a 1.
<code>a</code>	Numérico. Número inicial de indivíduos (abundância) por espécie. Deve ser maior ou igual a 1.
<code>nc</code>	Numérico. Número de ciclos a serem rodados. Deve ser maior ou igual a 0.
<code>dc</code>	Numérico. Número de mortes e nascimentos por ciclo. Deve ser maior que o número total de indivíduos presentes na comunidade, dado por $s*a$.
<code>nsim</code>	Numérico. Número de simulações (réplicas) a serem rodadas. Deve ser maior ou igual a 1.
<code>details</code>	Lógico. O array que contém a identidade de todos os indivíduos a cada ciclo (para todas as simulações) deve ser retornado ?

Details :

Em 2001, Hubbell organizou a Teoria Neutra da Biodiversidade, uma proposta para explicar a riqueza e a abundância de espécies de comunidades (e, por vezes, sua estrutura espacial) a partir de processos estocásticos de migração, especiação e sucessão de eventos de morte e nascimento de indivíduos. A função `hubbells.game()` utiliza apenas este último processo para determinar a dinâmica temporal da comunidade; desta forma, gera um modelo não-espacial. Na dinâmica, os indivíduos compõem um jogo de soma zero (o número de indivíduos total da comunidade não se altera) e sua reprodução é assexuada, já que não há formação de pares para gerar novos indivíduos. A característica herdada de parental para prole é sua identidade, isto é, a espécie a que pertence. Desta forma, ao final da dinâmica, podemos observar qual foi a variação temporal na riqueza e na abundância das espécies presentes na comunidade.

Value :

A função `hubbells.game()` retorna, no console, a média, utilizando-se todas as simulações rodadas, da riqueza final da comunidade. Isto é, calcula o número de espécies presentes na comunidade após os ciclos serem rodados para todas as simulações e retorna a média desses valores.

Na janela gráfica, a função retorna dois gráficos. O primeiro é a distribuição de abundância da comunidade em todas as simulações; a cada simulação é atribuída uma cor diferente. O segundo gráfico apresenta a variação da riqueza da comunidade ao longo dos ciclos rodados, novamente para todas as simulações.

Warning :

A função é interrompida e mensagens de erro são retornadas em casos de : ausência dos argumentos `s` e/ou `a` ; argumentos `s`, `a` e/ou `nsim` com valor menor ou igual a zero ; argumentos `nc` e/ou `dc` com valor menor que zero, argumento `dc` igual a zero quando argumento `nc` for maior que zero ; e argumento `dc` com valor maior ou igual ao número de indivíduos da comunidade.

Mensagens de aviso são retornadas em casos de : ausência do argumento `nc`, então será utilizado `nc` igual a zero ; ausência do argumento `dc`, então será utilizado `dc` igual a 1 ; ausência do argumento `nsim`, então será utilizado `nsim` igual a 1. Será enviada também uma mensagem de aviso quando o usuário determinar `nc` igual a zero, caso ele o tenha feito por engano. Ademais, é retornada uma mensagem de aviso que informa o usuário quanto à aplicação de um resíduo aleatório nos valores originais das variáveis abundância e riqueza nos gráficos gerados, para evitar que linhas ou pontos com mesmo valor de `x` e/ou `y` apareçam sobrepostos, dificultando a visualização das informações.

Author(s) :

Luísa Novara
luisanovara@gmail.com

References :

Hubbell, S.P. (2001) The Unified Neutral Theory of Biodiversity and Biogeography, *Princeton University Press*.

Rosindell, J.; Hubbell, S.P.; Etienne, R.S. (2011) The Unified Neutral Theory of Biodiversity and Biogeography at Age Ten. *Trends in Ecology & Evolution*. 26, 340–348.

See Also :

`sample()` para amostragem aleatória com ou sem reposição

Examples :

Exemplo 1 : Comunidade com riqueza 50 e abundância por espécie 3 submetida a 10 ciclos com 5 mortes e 5 nascimentos cada, replicado 10 vezes. Não foi solicitado o retorno do array com a identidade das espécies ao longo dos ciclos.

```
hubbells.game(50,3,10,5,10,F)
```

ou

```
hubbells.game(50,3,10,5,10) # o default do argumento details é FALSE
```

Observação : faça das duas formas e perceba que, ainda que as condições iniciais sejam as mesmas em ambos os casos, surgem resultados distintos, Isso ocorre porque a dinâmica neutra é estocástica.

Exemplo 2 : Comunidade com riqueza 20 e abundância por espécie 4 submetida a 1 ciclo com 15 mortes e 15 nascimentos, replicado 5 vezes.

```
hubbells.game(20,4,1,15,5,F)
```

Exemplo 3 : Mesma comunidade do exemplo anterior submetida à mesma dinâmica. Agora, é solicitado o retorno do array.

```
hubbells.game(20,4,1,15,5,T) # Observe que a identidade dos indivíduos muda do ciclo 0 (condição inicial) para o ciclo 1. Isso ocorreu porque foram selecionados aleatoriamente 15 indivíduos para serem eliminados e, dos indivíduos que restaram na comunidade, foram selecionados 15 para gerar um indivíduo de identidade equivalente a sua para ocupar os lugares vagos.
```

Exemplo 4 : Comunidade com riqueza 10 e abundância por espécie 5 submetida a nenhum ciclo, replicado 2 vezes. Note que, ainda que tenha sido definido um valor para o número de mortes e nascimentos por ciclo, ele não é utilizado.

```
hubbells.game(10,5,0,1,2) # os resultados finais reproduzem as condições
```

```
iniciais, já que foi definido valor zero para o número de ciclos rodados
```

```
hubbells.game(10,5,0,0,2) # esta linha de comando é funcionalmente equivalente à anterior
```

Entretanto, se for definido um valor maior que zero para o número de ciclos a serem rodados, deve ser definido também um valor maior que zero para o número de mortes e nascimentos por ciclo

```
hubbells.game(10,5,1,0,2) # erro na função
```

Exemplo 5 : Comunidade com riqueza 5 e abundância por espécie 2 submetida a 3 ciclos com 12 mortes e 12 nascimentos cada, com 1 réplica. A função é interrompida, já que o número de mortes/nascimentos por ciclo é maior do que o número total de indivíduos da comunidade.

```
hubbells.game(5,2,3,12,1) # dessa forma, a comunidade seria levada à extinção no primeiro ciclo
```

```
hubbells.game(5,2,3,8,1) # diminuindo o número de mortes por ciclo para um valor menor que o número total de indivíduos (no caso, 10), a função é executada corretamente
```