Quantificação da Diversidade Biológica

Prática 1 : Distribuições de Abundâncias

Paulo Inácio Prado e Adriana Martini

13 de novembro de 2010

Nestes exercícios vamos simular algumas comunidades com distribuições de abundâncias (SADs[1]) conhecidas, e ajustar alguns modelos teóricos pelo método da máxima verossimilhança. Como sabemos de antemão qual a SAD "real", podemos avaliar a eficácia do método de ajuste.

Em seguida repetiremos o mesmo procedimento para amostras tomadas destas comunidades, para avaliar o efeito da amostragem sobre a qualidade destes ajustes.

1 Preparando o R

Utilizaremos o ambiente estatístico R (R Development Core Team, 2010) [2], com algumas funções que preparamos para esta prática. Você receberá arquivos com estas funções, e também um arquivo com todos os comandos desta prática (`pratica1.r`).

Para executar este exercício: Você pode fazer este exercício mesmo que não conheça o R. Se você usa WINDOWS, abra o arquivo de comandos da prática no editor de *scripts*, com as opções `Arquivos > Abrir > Script` no menu principal da interface do R. Organize as janelas para ver tanto o *console* do R como a janela com o *script*, posicione o cursor em uma linha de comando e pressione `CTRL-R` para enviar o comando ao R . A idéia é ter o código aberto em um editor para programação, e enviar os comandos um a um para o RḢá muitos outros bons editores para isto, como o Emacs [3] com o módulos ESS [4] para todos os sistemas operacionais, e o Tinn-R [5] para Windows. Mais

[1] Species Abundance Distributions

[2] `http://www.r-project.org`

[3] `http://www.gnu.org/software/emacs/`

[4] `http://ess.r-project.org/`

[5] `http://www.sciviews.org/Tinn-R/`

detalhes sobre esta maneira de trabalhar no R em `http://ecologia.ib.usp.br/bie5782/doku.php?id=bie5782:02_tutoriais:tut1`.

Precisaremos também de alguns pacotes (*libraries*) que não fazem da instalação básica do R. Para começar, verifique se você já tem estes pacotes em sua instalação:

```
> lista <- installed.packages()
> lista[rownames(lista)=="vegan"|rownames(lista)=="bbmle"|
+       rownames(lista)=="poilog",c(1,3)]

       Package  Version
bbmle  "bbmle"  "0.9.5.1"
poilog "poilog" "0.4"
vegan  "vegan"  "1.17-4"
bbmle  "bbmle"  "0.9.5.1"
vegan  "vegan"  "1.17-4"
```

Se o resultado não for a lista dos três pacotes acima, instale-os a partir da internet com os comandos

```
> install.packages("vegan")
> install.packages("bbmle")
> install.packages("poilog")
```

E em seguida carregue-os na sua seção de R com:

```
> library(vegan)
> library(bbmle)
> library(poilog)
> library(MASS)
```

Por fim carregue os arquivos `funcoes_logser.r`, `funcoes_lognormal.r` e `rankplots.r` no seu diretório de trabalho [6]:

```
> source("funcoes_logser.r")
> source("funcoes_lognormal.r")
> source("funcoes_niche_part.r")
> source("rankplots.r")
```

[6] verifique qual é o diretório de trabalho atual com o comando `getwd()`

2 Simulação de Comunidades

2.1 Série Logarítmica

Caswell (1976) propôs um modelo muito simples de dinâmica neutra de comunidades, apenas com nascimentos, mortes e chegadas de colonizadores a intervalos discretos de tempo. O modelo estabelece que:

- A cada intervalo de tempo chegam em média ν indivíduos colonizadores, cada um de uma espécie que não existe na comunidade;
- As probabilidades *per capita* de produzir um filhote (λ) e morrer (μ) são constantes e iguais (ou seja, a taxa média de incremento populacional é zero);
- Estas probabilidades são as mesmas para todas as espécies (condição de neutralidade).

Caswell mostrou também que, dado tempo suficiente, esta comunidade teórica converge para uma distribuição de abundâncias de série logarítmica, em que o número esperado de espécies com n indivíduos é

$$E[n] = \frac{\alpha X^n}{n} \tag{1}$$

Sendo $\alpha = \nu/\lambda$. O parâmetro X é uma função de α e do total de indivíduos na comunidade, N:

$$X = \frac{N}{N+\alpha} \tag{2}$$

Use a função `caswell` para rodar este modelo nulo e guardar as abundâncias das espécies após 1000 intervalos de tempo:

```
> set.seed(51)
> ##Gera abundancias de especies com o modelo neutro de Caswell
> com1 <- caswell(nu=5,lambda=0.1,tmax=1000)
```

Inspecione o gráfico de ranque-abundâncias com as linhas previstas pelos modelo de série logarítmica, lognormal, geométrica e *brokenstick*:

```
> radplot(com1,lwd=1.5)
```

Compare o valor teórico de α com o obtido ajustando-se a série logarítmica aos dados, com a função `fit.logser`:

```
> 5/0.1

[1] 50

> ## Alfa obtido pelo ajuste a logserie
> com1.alfa <- fit.logser(com1)$summary["alfa"]
> com1.alfa

    alfa
53.85927
```

Com o que vimos até aqui a série logarítmica parece uma boa descrição da distribuição de abundâncias, como deduzido por Caswell. Para prosseguir, vamos também comparar as abundâncias previstas pela série logarítmica com os valores observados. Para isto tabulamos o número de espécies em cada classe de abundância e acrescentamos à tabela as probabilidades que o modelo de série logarítima atribui a cada abundância:

```
> com1.N <- sum(com1)
> com1.N

[1] 5055

> ##Numero de especies da comunidade
> com1.S <- length(com1)
> com1.S

[1] 245

> ## Uma tabela com N de individuos por abundancia
> com1.tab <- as.data.frame(table(com1))
> names(com1.tab)[1] <- "Abund"
> com1.tab$Abund <- as.integer(as.character(com1.tab$Abund))
> ##inspecionando os valores
> com1.tab

   Abund Freq
1      1   52
2      2   22
3      3   16
4      4   21
5      5    7
```

6	6	8
7	7	10
8	8	8
9	9	4
10	10	5
11	11	7
12	12	5
13	13	3
14	14	6
15	15	2
16	16	2
17	17	2
18	18	1
19	19	4
20	20	1
21	21	1
22	22	1
23	23	2
24	24	1
25	26	1
26	27	3
27	28	2
28	29	3
29	30	1
30	31	2
31	32	2
32	35	1
33	39	1
34	40	1
35	41	4
36	42	1
37	43	1
38	47	1
39	49	1
40	53	1
41	54	2
42	56	2
43	57	1
44	58	1
45	61	2

```
46    63    1
47    68    2
48    75    2
49    80    1
50    95    1
51   108    1
52   111    1
53   130    1
54   134    1
55   137    1
56   145    1
57   157    1
58   165    1
59   169    1
60   182    1
61   188    1
62   198    1
63   218    1
```

```
> ## Probabilidades atribuidas pelo modelo a cada abundancia observada
> com1.tab$p.ls <- dlr(com1.tab$Abund,N=com1.N,alfa=com1.alfa)
> ##inspecionando primeiros valores
> head(com1.tab)
```

```
  Abund Freq       p.ls
1     1   52 0.21735064
2     2   22 0.10752963
3     3   16 0.07093068
4     4   21 0.05263718
5     5    7 0.04166581
6     6    8 0.03435546
```

Vamos agora calcular os números esperados de espécies para cada valor de abundância. Para isto, multiplicamos cada probabilidade pelo total de espécies:

```
> com1.tab$Pred.Ab.ls <- com1.tab$p.ls*com1.S
> ##inpecionando
> head(com1.tab)
```

```
  Abund Freq       p.ls Pred.Ab.ls
1     1   52 0.21735064  53.250907
```

```
2     2   22 0.10752963  26.344759
3     3   16 0.07093068  17.378016
4     4   21 0.05263718  12.896108
5     5    7 0.04166581  10.208123
6     6    8 0.03435546   8.417088
```

A série logarítmica parece um descrição adequada, mas temos que avaliar se de fato é a melhor. O gráfico sugere que a lognormal pode ser uma descrição plausível. Comparamos a plausibilidade de dois modelos estatísticos com a **função de log-verossimilhança**, que é simplesmente a soma dos logarítmos das probabilidades atribuídas pelo modelo a cada observação, que é a abundância de cada espécie. Como já fizemos o ajuste do modelo e temos uma função que atribui probabilidades a cada valor de abundância [7], podemos calcular a verossimilhança. Para isto calculamos o logaritmo das probabilidades atribuídas a cada abundância, multiplicamos pelo número de observações (espécies) com esta abundância, e somamos todos estes valores:

```
> com1.LL.ls <- sum(log(com1.tab$p.ls)*com1.tab$Freq)
```

Em seguida, repetimos os mesmos procedimentos para lognormal, que ajustamos com a função `lnormt.fit`. Com isto obtemos os parâmetros do modelo lognormal para estes dados, que usamos para calcular as probabilidades atribuídas por este modelo a cada abundância. Em seguida calculamos os esperados e a verossimilhança:

```
> com1.pln <- coef(lnormt.fit(com1))
> com1.pln

       m        s
1.557870 1.763289

> ## Probabilidades atribuidas a cada abundancia pela lognormal
> com1.tab$p.ln <- plnormt.ab(com1.tab$Abund,mu=com1.pln[1],sigma=com1.pln[2])
> ## N previsto de especies em cada abundancia
> com1.tab$Pred.Ab.ln <- com1.tab$p.ln*com1.S
> ##inspecionando primeiras linhas da tabela
> head(com1.tab)

  Abund Freq       p.ls Pred.Ab.ls       p.ln Pred.Ab.ln
1     1   52 0.21735064  53.250907 0.17331011   42.46098
```

[7] esta é a função de distribuição de probabilidades do modelo

```
2     2   22 0.10752963  26.344759 0.11264696   27.59850
3     3   16 0.07093068  17.378016 0.08155581   19.98117
4     4   21 0.05263718  12.896108 0.06285930   15.40053
5     5    7 0.04166581  10.208123 0.05045025   12.36031
6     6    8 0.03435546   8.417088 0.04166340   10.20753
```

```
> ## Log-verossimilhanca do modelo log-normal
> com1.LL.ln <- sum(log(com1.tab$p.ln)*com1.tab$Freq)
```

A diferença entre as log-verossimilhanças é uma medida de plausibilidade relativa, em escala logarítmica. No caso, podemos concluir que a série logarítimica é uma descrição $e^{3,5}$ mais plausível que a lognormal:

```
> com1.LL.ls
```

```
[1] -891.5101
```

```
> com1.LL.ln
```

```
[1] -895.0407
```

```
> ## Razao de verossimilhancas
> com1.LL.ls-com1.LL.ln
```

```
[1] 3.530564
```

```
> ## Razao de verossimilhanças em escala aritmética
> exp(com1.LL.ls-com1.LL.ln)
```

```
[1] 34.14320
```

Esta comparação ainda não é justa, pois a lognormal tem dois parâmetros e a série logarítmica apenas um. Nestes casos, usamos o **Critério de Informação de Akaike** (AIC), que penaliza modelos mais complicados, ou seja, com mais parâmetros. O AIC é simplesmente o dobro da log-verossimilhança negativa, somado a duas vezes o número de parâmetros:

```
> -2*com1.LL.ls+2
```

```
[1] 1785.020
```

```
> ## lognormal
> -2*com1.LL.ln+4
```

```
[1] 1794.081
```

O AIC é uma estimativa da distância de cada modelo ao modelo correto. Portanto, o modelo de menor AIC é a melhor aproximação. Modelos com uma diferença de AIC de dois ou menos são igualmente plausíveis.

2.2 Lognormal e Brokenstick

A função `caswell.ln` simula uma variante do modelo de Caswell descrito acima, em que não há colonizadores, e as probabilidades de nascimentos e mortes podem ser diferentes. Vamos simular uma comunidade que começa com 200 espécies, cada uma com 20 indivíduos, e probabilidade de nascimentos ligeiramente maior do que a de mortes. Geramos um vetor de abundâncias das espécies com:

```
> set.seed(42)
> com2 <- caswell.ln(S0=200,n0=20,pbirth=0.105,pdeath=0.1,tmax=250)
```

Inspecionamos visualmente os ajustes dos modelos:

```
> radplot(com2)
```

As distribuições *Brokenstick* e lognormal parecem ser as melhores descrições para estes dados. Embora a série logarítmica não pareça uma boa alternativa, vamos ajustá-la também, e comparar os resultados. Na seção anterior fizemos os cáculos passo a passo para compreendê-los. Agora vamos usar as funções que já fazem o ajuste calculam as log-verossimilhanças, que são `fit.logser` e `lnormt.fit`:

```
> com2.fit.ls <- fit.logser(com2)
> com2.fit.ln <- lnormt.fit(com2)
```

Para o ajuste da distribuição *Brokenstick* basta termos o número de espécies e de indivíduos na amostra, portanto apenas calculamos a sua log-verossimilhança e seu AIC:

```
> com2.N <- sum(com2)
> com2.S <- length(com2)
> com2.LL.bs <- sum(log(rad.bs(com2,com2.N,com2.S)/com2.S))
> com2.AIC.bs <- -2*com2.LL.bs
```

E agora comparamos as log-verossimilhanças e AICs:

```
> com2.fit.ls$summary
```

```
            N             S          alfa             X        loglik
12926.0000000   145.0000000    22.8900238     0.9982323  -853.0968375
          AIC
 1708.1936750
```

```
> ## lognormal
> c(N=com2.N,S=com2.S,coef(com2.fit.ln),
+   loglik=logLik(com2.fit.ln),AIC=AIC(com2.fit.ln))

           N            S            m            s       loglik          AIC
12926.000000   145.000000     3.878479     1.253601  -800.821791  1605.643582

> ## brokenstick
> c(N=com2.N,S=com2.S,coef(com2.fit.ln),
+   loglik=com2.LL.bs,AIC=com2.AIC.bs)

           N            S            m            s       loglik          AIC
12926.000000   145.000000     3.878479     1.253601  -796.164437  1592.328875
```

E comparamos os dois modelos com maiores verossimilhanças:

```
> as.numeric(com2.LL.bs-logLik(com2.fit.ln))

[1] 4.657353

> ## delta-AIC brokenstick x lognormal
> AIC(com2.fit.ln)-com2.AIC.bs

[1] 13.31471
```

Pergunta 1: Qual o modelo mais plausível neste caso? Proponha uma explicação.

3 Efeito da Amostragem

3.1 Série Logarítmica

Na seção anterior tivemos acesso à toda a comunidade, mas na vida real em geral temos apenas amostras das comunidades. Como a msotragem afeta o ajuste do modelos? Podemos explorar tomando ao acaso uma fração dos indivíduos das comunidades que criamos. Vamos começar com a primeira, tomando uma amostra de 10% de seus indivíduos:

```
> com1.ind <- rep(1:length(com1),com1)
> ## uma amostra de 10% da comunidade
> set.seed(42)
> com1.s <- table(sample(com1.ind,size=round(length(com1.ind)*0.1)))
```

Agora comparamos os gráficos de ranque-abundância da comunidade e desta amostra:

```
> w.plot(com1)
> lines(rad.logser(com1))
> points.w.plot(com1.s,col="red")
```

Sabemos que a comunidade segue uma série logarítmica. Vamos ajustar este modelo aos dados da comunidade e da amostra, e comparar os valores de seu único parâmetro, α:

```
> com1.fit.ls <- fit.logser(com1)
> com1s.fit.ls <- fit.logser(com1.s)
> com1.fit.ls$summary
```

```
           N            S         alfa            X       loglik          AIC
5055.0000000  245.0000000   53.8592720    0.9894577 -891.5101421 1785.0202842
```

```
> com1s.fit.ls$summary
```

```
           N            S         alfa            X       loglik          AIC
 506.0000000  126.0000000   53.7736567    0.9039368 -271.1872245  544.3744489
```

Acrescentamos ao gráfico a linha prevista usando o α da comunidade, em vermelho:

```
> lines(rad.logser(com1.s),col="red",lwd=2)
```

E agora a mesma linha, para o alfa obtido com o ajuste aos dados da amostra:

```
> lines(rad.logser(com1.s,alfa=com1s.fit.ls$summary["alfa"]),col="blue", lty=2)
```

3.2 Lognormal

Vamos repetir o mesmo procedimento para uma amostra de 10% da segunda comunidade simulada na seção anterior:

```
> com2.ind <- rep(1:length(com2),com2)
> ## uma amostra de 10% da comunidade
> set.seed(42)
> com2.s <- table(sample(com2.ind,size=round(length(com2.ind)*0.1)))
```

Compare o valor dos parâmetros estimados para a logormal ajustada à comunidade e à amostra:

```
> com2s.fit.ln <- lnormt.fit(com2.s)
> ## comparando as estimativas dos parametros
> ## Comunidade
> coef(com2.fit.ln)

       m        s
3.878479 1.253601

> ## Amostra
> coef(com2s.fit.ln)

       m        s
1.821194 1.049333
```

Os dois parâmetros da lognormal são afetados pela amostragem. O parâmetro μ está relacionado principalmente ao total de indivíduos observados. Se conhecemos a fração dos indivíduos da comunidade que foram amostrados a, uma correção simples para estimar o valor do μ na comunidade a partir do estimado na amostra (m) é:

$$\hat{\mu} \; = \; m - ln(a) \tag{3}$$

Como aqui conhecemos a fração de indivíduos amostrados, que é de 0.1, podemos aplicar esta correção ao valor estimado pela amostra, e obtemos um valor bem próximo ao valor de μ na comunidade:

```
> as.numeric(coef(com2s.fit.ln)[1])-log(0.1)

[1] 4.123779
```

Mas o parâmetro mais interessante é σ, que está relacionado à dominância. Não há correção proposta para ele e em nossa simulação ele foi subestimado na amostra. Para avaliar se esta subestimativa causa um desvio importante nos valores previstos, fazemos um um gráfico de ranque-abundância dos valores de abundância na amostra e sobrepomos a linha do previsto pelos valores ajustados para a amostra e com valor de σ da comunidade:

```
> w.plot(com2.s)
> lines(rad.lnormt(com2.s),col="blue")
> lines(rad.lnormt(com2.s,sigma=coef(com2.fit.ln)[2]), col="red")
> legend(x="topright",
+        legend=c("Amostra",expression(paste(sigma, " comunidade"))),
+        lty=1,col=c("blue","red"))
```

Pergunta 2: Quais suas conclusões? Justifique brevemente.

Referências

Caswell, H., 1976. Community structure: a neutral model analysis. *Ecological Monographs* pages 327–354.

R Development Core Team, 2010. R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0.