

Pedro Henrique Imenez Silva



Doutorando pelo programa de Fisiologia Humana do Instituto de Ciências Biomédicas na área de Fisiologia Renal

exec

Propostas de função

A) A primeira proposta não está relacionada com o meu projeto de doutorado, mas está relacionado a outro assunto que gosto que é educação. Alguns anos atrás, durante a licenciatura, um colega de classe e eu distribuimos um questionário para alunos de ensino fundamental e médio com o intuito de avaliar os conceitos e opiniões dos alunos sobre diferentes tipos de fontes de energia (solar, eólica, nuclear, etc.). Após uma varredura inicial na qual analisamos parâmetros simples, como média e mediana das respostas, passamos todos os dados para tabelas de excel e atrelamos respostas de determinadas questões com outras. Para isto os dados eram inseridos nas planilhas em forma binária, de modo que valores de 0 e 1 representassem algum “teste lógico” com outras questões. Estas análises nos permitiram tirar algumas conclusões bem interessantes, porém foi bem cansativo, e de certa forma bem rústico. Assim, pretendo com esta função propor um meio de analisar correlações em um ambiente como o de questionários, onde se pretende menos avaliar padrões gerais de opinião e mais tentar entender se os indivíduos formam padrões conceituais sobre determinado tema. Este trabalho se baseará em resgatar os dados originais e formatá-los de modo adequado para a análise em R. Após, retornarei às questões originais que motivaram o trabalho para tentar entender através de testes lógicos se um indivíduo que tem preferência por uma resposta de uma questão x escolhe mais determinadas respostas em questões y e z. Como exemplo poderíamos imaginar um cenário no qual um adolescente que acessa menos a internet poderia ter a propensão a escolher determinadas fontes energéticas como prioritária para o mundo, mas na posição de presidente do país abandonaria a sua opinião original e escolheria outra fonte. A ideia é brincar com as diferentes relações das respostas. Dificuldades: irá exigir uma boa arqueologia dos dados. Tenho quase tudo documentado do processo, mas algo pode faltar. A minha ideia é identificar a viabilidade da análise o quanto antes para decidir se devo ou não ir para o plano B.

B) RNAs mensageiros (RNAm) possuem como qualquer outro componente do nosso organismo uma “meia-vida”. Os RNAm degradam em diversas e diferentes taxas, mas uma grande parte (até se imagina que a maioria) deles respeita uma taxa de decaimento exponencial. No início do doutorado eu me preocupei com possíveis modificações da taxa de decaimento do RNAm de uma proteína chamada NHE3 (do inglês, isoforma 3 do trocador sódio hidrogênio) causadas por dois tipos de estresses fisiológicos: acidose metabólica e acidose respiratória. Estas condições patofisiológicas poderiam mudar a estabilidade do RNAm-NHE3 e modificar sua taxa de decaimento. Para avaliar esta questão submeti linhagens celulares derivadas de túbulo proximal de rins de *Didelphis virginiana* (opossum, um gambá) às condições que simulassem os dois tipos de acidose por 48 horas. Ao final das primeiras 24h de tratamento eu interrompia farmacologicamente a produção de RNAm das células e coletava em diversos tempos o RNA total. Após manobras experimentais de biologia molecular eu quantificava o RNAm-NHE3 dos tempos coletados observando o nível de RNAm sem o efeito da síntese de novas fitas, ou seja apenas com o fator degradação atuando. Com esta função pretendo analisar desde os dados iniciais chamados “ct”s que permitem que eu deduza o nível de RNAm

presente na amostra no momento da coleta, até produzir as curvas de decaimento. Caso seja possível pretendo simular mudanças bioquímicas que resultariam em modificações no decaimento do RNAm (na realidade na função do decaimento) e observar os resultados graficamente.

Comentários das propostas (Leo)

As duas propostas parecem interessantes e factíveis. Particularmente gosto mais da segunda, pois envolve uma gama mais diversificada de tarefas: manipulação da tabela de entrada, aplicação de alguma equação (ct para RNAm presente), ajuste de uma função aos dados (curvas de decaimento), plotagem de um gráfico, etc. Isto também é um problema recorrente na biologia: ajustar curvas de diversas naturezas aos dados. Um desafio adicional seria ajustar diferentes curvas de decaimento e selecioná-las por algum critério específico de ajuste (p.ex. AIC). Mas, é claro, não precisa “conquistar o mundo” como o “cérebro”. 😊 Vai até onde o prazo para entregar o exercício permitir.

Senti falta de uma descrição mais detalhada das tabelas/dados de entrada e saída (inputs e outputs da função). Além disto seria interessante tentar quebrar a “grande tarefa” em tarefas menores, explicitando-as. Isto ajudará o leitor a avaliar a tua função e o tamanho do desafio envolvido, além de te ajudar a organizar o trabalho de construção da mesma.

Resposta aos comentários

Obrigado pelos comentários, todos eles foram aceitos. Abaixo breves comentários sobre cada ponto.

- A proposta B foi escolhida para a função.
- Foram feitos apenas dois tipos de regressões (linear e exponencial negativa de uma fase). Havia interesse em adicionar uma exponencial negativa de duas fases, mas não consegui transmitir para os códigos a linguagem matemática, portanto não foi possível executá-la. Outros tipos de regressões foram cogitados, mas para alguns mais fáceis não existiam razões teóricas para usar e para outras eu acabei deixando para caso sobrasse tempo. Infelizmente não consegui implementá-las há tempo.
- Utilizei o AIC, porém confesso que o entendimento do parâmetro foi parcial. Aparentemente ele funcionou, mas sem os conhecimentos dos pressupostos e um pouco de verosimilhança ou bayesiana (BIC) a utilização se torna um pouco irresponsável. Deixarei uma nota no “help”
- A descrição dos dados de entrada e saída ficarão mais claros no comentários feitos em meio a função e no artigo help inserido aqui. Além disso, adicionarei uma explicação mais genérica sobre a função logo abaixo.
- Não tentei dominar o mundo, mas foi uma ótima oportunidade para treinar o uso do R. Infelizmente terei que entregar antes do prazo final, o que não me possibilitará ter algumas noites adicionais para tentar fazer o que tenho feito em todas as últimas noites 😊.

Comentários extras sobre a função

PCR é a sigla em inglês para a Reação em Cadeia da Polimerase (Polymerase Chain Reaction), técnica de biologia molecular que permite amplificar o número de cópias de uma determinada sequência de

DNA. Este método, desenvolvido por Kary Mullis em 1983, consiste em expor a amostra de DNA a uma temperatura que irá separar a sua dupla fita, para que assim, pequenas sequências de nucleotídeos chamadas iniciadores (primers) pareiem com a sua sequência complementar. Os iniciadores aleatoriamente fazem uma espécie de busca das sequências que lhe são complementares, identificando as moléculas alvo no meio de muitas outras moléculas presentes na amostra. Por fim, uma enzima chamada polimerase promove a extensão do DNA. Estas fases se repetem em diversos ciclos, possibilitando que um grande número de sequências seja criado.

Os avanços do PCR quantitativo recaíam sobre a quantificação de DNA, porém elas não eram aplicáveis ao RNA. Poucos anos após a invenção do PCR, em 1987, Powell e colaboradores descreveram um método eficaz na quantificação de RNA. A técnica consistia em tornar um RNA fita única em um híbrido de RNA com uma fita de DNA complementar, o cDNA, pela atividade da enzima transcriptase reversa (RT). Assim, foi possível utilizar o PCR quantitativo sobre o material genético que originalmente era RNA. Este é o RT-PCR. Por último, na década de 1990 o tempo real foi adicionado aos PCRs. A ideia do PCR em tempo real é observar a amplificação exponencial do DNA durante a reação de PCR e com isso inferir qual era a quantidade absoluta ou relativa de material genético no início da reação. No PCR em tempo real o produto de PCR gera (de modos diferentes dependendo do sistema empregado) uma fluorescência que é medida a cada ciclo. O limiar de fluorescência detectável será atingido com menor número de ciclos quanto maior o número inicial de moléculas. O que se detecta, portanto, é ciclo (ct) no qual a fluorescência atinge o limiar, não é mais a quantidade final de produto na reação.

Com a função `rna.decay()` apresentada nesta página é possível calcular a partir de uma tabela de valores de ct, os níveis de RNAm em diferentes tempos ou condições. A função retorna:

1. as regressões exponenciais e lineares feitas sobre os dados, e seus coeficientes;
2. teste de AIC para as regressões;
3. gráficos das regressões.

obs: este texto foi adaptado do texto “PCR quantitativo em tempo real” escrito por Pedro Henrique Imenez Silva e Nancy Amaral Rebouças em *Praticando e discutindo fisiologia : IX Curso de Verão / coordenação de Thiago dos Santos Moreira e pós-graduandos do Departamento de Fisiologia e Biofísica*. São Paulo : ICB/BMB, 2011. 77f. : il.

Página de Ajuda

```
rna.decay          package:nenhum          R Documentation
      Regressão exponencial negativa de taxa de decaimento de RNAm
Description:
```

Função que retorna coeficientes das funções exponenciais negativas de uma fase e das funções lineares descritas pelas regressões dos valores de ct (cycle threshold) obtidos de uma PCR em tempo real(real time polymerase chain reaction).

Os dados de níveis de RNAm obtidos a partir dos ct e as curvas das regressões são fornecidos graficamente.

Usage:

```
rna.decay(ct, t0, t1, t2, t3, t4, t5, aed=TRUE)
```

Arguments:

ct - dataframe com os valores de ct
t0,...,t5 - tempo após interrupção da transcrição (t0) no qual o RNA foi extraído
aed - TRUE ou FALSE, define o tipo de resultado gráfico resultante.

Details:

ct: valores brutos de ct obtidos do pcr em tempo real.
O dataframe deve possuir obrigatoriamente 5 colunas:
Coluna 1: denominações "t0", "t1", "t2"... "t5" para cada linha. Todas devem ser preenchidas, mas não é necessário ocupar todos os tempos, por exemplo pode se utilizar de t0 a t4, sem necessidade de utilizar t5.
Coluna 2 e 3: cada coluna deve receber os ct de um grupo. Caso a análise não seja comparativa e se utilize apenas um grupo, somente a coluna 2 dois deve ser preenchida com os valores de ct, e a coluna 3 deve ser preenchida com zeros em todas as linhas.
Colunas 4 e 5: cada coluna deve receber os ct obtidos do PCR que utilizou iniciadores (primers) específicos para o controle endógeno (normalizador de quantidades). Coluna 4 é respectiva à coluna 2 e a coluna 5 à coluna 3. Caso não queira utilizar normalização dos dados, as colunas 4 e 5 devem ser preenchidas com zeros em todas as linhas.
t0...t5: tempo em minutos após interrupção da transcrição (t0) no qual o RNA foi extraído. Caso não sejam utilizados todos os tempos, preencha os valores necessários e os excedentes podem ser preenchidos com qualquer valor superior ao último estabelecido. Por exemplo, se t4 foi o último utilizado, preencha t5 com qualquer valor superior a t4. Porém há restrições na não utilização de todos os tempos (ver argumento aed=FALSE).

aed: como padrão é utilizado aed=TRUE. O aed=TRUE retorna uma janela gráfica com 3 gráficos de regressão linear e exponencial negativa. Além disso, esta condição garante que se utilize menor variedade de tempos (t0 a t5) sem prejuízo nos resultados. Adicionalmente o aed=TRUE apresenta os valores de AIC das regressões efetuadas e os coeficientes das funções. aed=FALSE retorna uma janela gráfica com médias e desvios dos dados e a regressão exponencial de cada grupo.

Values:

Se aed=TRUE rna.decay() retorna três gráficos com os dados de nível de RNAm plotados e com regressões lineares ou exponenciais negativas de uma fase.
Se aed=TRUE rna.decay() retorna valores de AIC das regressões lineares e exponenciais negativas de uma fase, além dos coeficientes das funções que descrevem as regressões.
Se aed=FALSE rna.decay() retorna um gráfico com médias e desvios de cada grupo em cada tempo e suas respectivas regressões exponenciais negativas de uma fase.

Warning:

As regressões assim como o teste de AIC possuem pressupostos que não são testados pelas funções.

A regressão exponencial negativa utiliza a função $N(t) = N(t_0) \cdot \exp(-\lambda \cdot t)$, sendo $N(t)$ o nível de RNAm no tempo t , $N(t_0)$ o nível de RNAm no tempo t_0 e λ é a constante de decaimento da exponencial negativa.

Author(s):

Pedro Henrique Imenez Silva
ph@icb.usp.br
peagasilva@gmail.com

References:

Livak KJ, Schmittgen TD (2001) Analysis of relative gene expression data using real-time quantitative PCR and the $2^{-\Delta\Delta C(T)}$ Method. *Methods* 25:402–408

Silva, PH, Girardi AC, Neri, EA, Rebouças NA (2012) Distinct mechanisms underlie adaptation of proximal tubule Na^+/H^+ exchanger isoform 3 in response to chronic metabolic and respiratory acidosis. *Pflugers Arch* 463:703-714

See also:

'AIC' e 'nls' do pacote stats

Exemples:

Baixe a tabela "rnaexemplo.csv" em
http://ecologia.ib.usp.br/bie5782/doku.php?id=bie5782:01_curso_atual:alunos:trabalho_final:ph:exec

```
ct <- read.csv("rnaexemplo.csv", sep=";", header=T, dec=",")
rna.decay(ct, t0=0, t1=180, t2=360, t3=540, t4=720, t5=(60*24), aed=F)
rna.decay(ct[1:20,], t0=0, t1=180, t2=360, t3=540, t4=720, t5=(60*24), aed=T)
```

Código da Função

```
rna.decay <- function(ct, t0=x, t1=y, t2=z, t3=a, t4=b, t5=c, aed=TRUE)
{
  names(ct) <- c("tempo", "g1", "g2", "g1.end", "g2.end")
  delta.geral.g1 <- ct$g1 - ct$g1.end # esta analise exige a normalizacao
dos dados por um controle endogeno que nao deve variar de acordo com o
tratamento
  delta.geral.g2 <- ct$g2 - ct$g2.end
  media.g1.t0 <- mean(delta.geral.g1[ct$tempo=="t0"], na.rm=TRUE)
  delta.delta.g1 <- delta.geral.g1 - media.g1.t0
  delta.delta.g2 <- delta.geral.g2 - media.g1.t0 # análise pode ser feita
em relação ao próprio tempo zero ou ao do grupo 1 (controle). Optei por
```

```
fazer pelo controle, nada impede de mudar a funcao criando um "media.g2.t0"
dois.a.delta.g1 <- 2^(-delta.delta.g1) # aqui finaliza o calculo da
quantidade inicial de mRNA a partir do valor de ct (2^(-delta.delta(ct)))
dois.a.delta.g2 <- 2^(-delta.delta.g2) # esta abordagem tem uma serie de
pressupostos que o usuario tem que estar ciente (normalização, razao pelo
controle, analise comparativa de dois grupos).

#Não considero isto como restricao, mesmo porque a função tem
frixibilidade para aceitar diferentes padrões de tabelas, eh mais questão de
ficar claro para o usuário a necessidade de preparar seus experimentos desde
o início com controles adequados e que a abordagem é comparativa
tempo.each <- rep(c(t0,t1,t2,t3,t4,t5), c(length(ct$g1[ct$tempo=="t0"]),
length(ct$g1[ct$tempo=="t1"]),
length(ct$g1[ct$tempo=="t2"]),length(ct$g1[ct$tempo=="t3"]),length(ct$g1[ct$
tempo=="t4"]), length(ct$g1[ct$tempo=="t5"])))
if (aed==TRUE)
{
  par(mfrow=c(1,3))
  plot(dois.a.delta.g1~tempo.each, main="Ajuste linear e exponencial
negativo de G1", xlab="tempo (min)", ylab="Nível relativo de mRNA")
  abline(ct.lm.g1 <- lm(dois.a.delta.g1~tempo.each)) #inicio dos
fittings
  coef.lm.g1 <- coefficients(ct.lm.g1)
  lambda.g1<- -log(dois.a.delta.g1)/ tempo.each # inicio do fitting
exponencial negativo de uma fase.
  lambda.g1a <-lambda.g1
  lambda.g1a[which(lambda.g1== -Inf|lambda.g1==Inf)]= NA
  lambda.start.g1 <- mean(lambda.g1a, na.rm=T) # o calculo dessa parte
existe apenas para fornecer o valor necessario de "start" para a funcao
"nls" abaixo
  fit.exp.g1<- nls(dois.a.delta.g1~a*exp(b*tempo.each), start=c("a"=1,
"b"=lambda.start.g1)) # atencao eh necessaria em toda esta fase, pois nao
foi utilizado um fitting pronto como no caso do lm.
  #o processo passa por uma funcao exponencial que pode ser
contestada. Caso se enxergue o processo de outra maneira aqui deve ser
modificado (assim como o calculo do lambda.start)
  coef.fit.g1 <- coefficients(fit.exp.g1)
  curve(coef.fit.g1[1]*exp(coef.fit.g1[2]*x), add=T, col="blue")
  plot(dois.a.delta.g2~tempo.each, main="Ajuste linear e exponencial
negativo de G2", xlab="tempo (min)", ylab="Nível relativo de mRNA")
  abline(ct.lm.g2 <- lm(dois.a.delta.g2~tempo.each))
  coef.lm.g2 <- coefficients(ct.lm.g2)
  lambda.g2<- -log(dois.a.delta.g2)/ tempo.each
  lambda.g2a <-lambda.g2
  lambda.g2a[which(lambda.g2== -Inf|lambda.g2==Inf)]= NA
  lambda.start.g2 <- mean(lambda.g2a, na.rm=T)
  fit.exp.g2<- nls(dois.a.delta.g2~a*exp(b*tempo.each), start=c("a"=1,
"b"=lambda.start.g2))
  coef.fit.g2 <- coefficients(fit.exp.g2)
  curve(coef.fit.g2[1]*exp(coef.fit.g2[2]*x), add=T, col="red")
```

```

dados.geral <- c(dois.a.delta.g1, dois.a.delta.g2)
plot(dados.geral~rep(tempo.each, 2),
col=rep(c(4,2),c(length(dois.a.delta.g1),length(dois.a.delta.g2))),
main="Ajuste exponencial de G1 (azul) e G2 (vermelho)", xlab="tempo
(min)", ylab="Nível relativo de mRNA")
curve(coef.fit.g1[1]*exp(coef.fit.g1[2]*x), add=T, col="blue")
curve(coef.fit.g2[1]*exp(coef.fit.g2[2]*x), add=T, col="red")
AIC.fits<-AIC(ct.lm.g1, ct.lm.g2, fit.exp.g1, fit.exp.g2) # O AIC
foi utilizado como recomendado, mas não o entendo direito
# Não me sinto confortável fazendo algo sem entender direito. De
qualquer modo, os menores valores batem com os melhores fits como esperado.
rownames(AIC.fits) <- c("lm(g1)", "lm(g2)", "nls(g1)", "nls(g4)")
lista.rna <- data.frame(coef.fit.g1, coef.fit.g2, coef.lm.g1,
coef.lm.g2)
colnames(lista.rna) <- c("Coeficientes nls(G1)", "Coeficientes
nls(G2)", "Coeficientes lm(G1)", "Coeficientes lm(G2)")
rownames(lista.rna) <- c("Intercepto", "lambda/inclinacao")
lista.rna.2 <- list(AIC.fits, lista.rna)
return(lista.rna.2)
}
else
{
tempo <-c(t0,t1,t2,t3,t4,t5)
delta.df <- data.frame(delta.geral.g1, delta.geral.g2, tempo.each)
delta.g1<- tapply(delta.df$delta.geral.g1, delta.df$tempo.each,
mean, na.rm=TRUE)
delta.g2<- tapply(delta.df$delta.geral.g2, delta.df$tempo.each,
mean, na.rm=TRUE)
deltas <-c(delta.g1, delta.g2)
deltas.geral <-deltas-delta.g1[1]
doisadeltas <- 2^(-deltas.geral)
plot(doisadeltas~rep(tempo,2),
pch=rep(c(16,17),c(length(tempo),length(tempo))), ylim=c(0,1.5),
main="Ajuste exponencial de G1 (círculo) e G2 (triângulo)", xlab="tempo
(min)", ylab="Nível relativo de mRNA", bty="l", tcl=0.3, xaxt="n")
axis(side=1, at=c(0,180,360,540,720, 900, 1080, 1260,1440), tcl=0.3)
deltas.df<- data.frame(dois.a.delta.g1, dois.a.delta.g2, tempo.each)
sd.g1 <- tapply(deltas.df$dois.a.delta.g1, tempo.each, sd,
na.rm=TRUE)
sd.g2 <- tapply(deltas.df$dois.a.delta.g2, tempo.each, sd,
na.rm=TRUE)
sd <- c(sd.g1, sd.g2)
sd.mais <- doisadeltas+sd
sd.menos <- doisadeltas-sd
sds <- c(sd.mais, sd.menos)
segments(x0=rep(tempo,2), y0=c(sd.mais), x1= (rep(tempo,2)),
yl=c(sd.menos))
lambda.gla <-lambda.g1<- -log(dois.a.delta.g1)/ tempo.each # esta
parte eh apenas uma repeticao do do que foi feito no "if". Para nao
atropelar a sequencia de ideias preferi repetir esta parte
lambda.gla[which(lambda.g1== -Inf|lambda.g1==Inf)]= NA

```

```
lambda.start.g1 <- mean(lambda.g1a, na.rm=T)
fit.exp.g1<- nls(dois.a.delta.g1~a*exp(b*tempo.each), start=c("a"=1,
"b"=lambda.start.g1))
coef.fit.g1 <- coefficients(fit.exp.g1)
lambda.g2a <- lambda.g2<- -log(dois.a.delta.g2)/ tempo.each
lambda.g2a[which(lambda.g2== -Inf|lambda.g2==Inf)]= NA
lambda.start.g2 <- mean(lambda.g2a, na.rm=T)
fit.exp.g2<- nls(dois.a.delta.g2~a*exp(b*tempo.each), start=c("a"=1,
"b"=lambda.start.g2))
coef.fit.g2 <- coefficients(fit.exp.g2)
curve(coef.fit.g1[1]*exp(coef.fit.g1[2]*x), add=T)
curve(coef.fit.g2[1]*exp(coef.fit.g2[2]*x), add=T)
}
```

Arquivo da Função

[rna_decay.r](#)

From:

<http://ecor.ib.usp.br/> - **ecoR**

Permanent link:

http://ecor.ib.usp.br/doku.php?id=05_curso_antigo:r2013:alunos:trabalho_final:ph:start

Last update: **2020/08/12 06:04**

