Luiz Gustavo Dufner de Almeida

Resumo de atividades

Aluno de doutorado do Departamento de Genética e Biologia Evolutiva do Instituto de Biociências-USP, com graduação e mestrado no próprio instituto. Atualmente desenvolvo estudo de mutações e alterações nas expressões dos genes *TSC1* e *TSC2* do complexo da esclerose tuberosa (TSC) utilizando *big data* de sequenciamento de próxima geração (NGS - *next generation sequencing*).

Curriculo Lattes

http://lattes.cnpq.br/6158194650895425

Dados para contato

Nome	Luiz Gustavo Dufner de Almeida
Projeto	Estudo mutacional e funcional dos genes 1 e 2 do complexo da esclerose tuberosa
e-mail 1	luiz.gustavo.almeida@usp.br
e-mail 2	lgustavo85@gmail.com
Telefone	(11)986427433

Exercícios

Exercício	Arquivo
1	exercicios_1.r
2	Somente NotaR
3	Somente NotaR
4	exercicios_4_caixetas.r, exercicios_4_cervejas.r, exercicios_4_eucaliptos.r, exercicios_4_mais_caixetas.r
5	exercicios_5_editando_parametros.r, exercicios_5_dois_graficos.r, exercicios_5_adivinhando_codigo.r
6	Somente NotaR
7a	exercicios_7_o_modelo_mais_simples_possivel.r
7b	exercicios_7b_galileu.r, exercicios_7b_massas_rescem_nascidos.r, exercicios_7b_babies_1.r
8	exercicios_8_ex1.r, exercicios_8_ex2.r
9	exercicios_9_2.r

Propostas de Trabalho Final

Proposta 1

Análise de qPCR pelo método ΔΔCt

A técnica de qPCR (*quantitative polymarase chain reaction*) é largamente utilizada na quantificação da expressão gênica ou na comparação de segmentos genômicos deletados ou duplicados. Essa técnica tem como base uma reação de PCR (*polymarase chain reaction*) utilizando-se um marcador fluorescente que permite detectar a luminescência da amostra a cada ciclo de duplicação do DNA em uma PCR.

A análise de qPCR consiste na comparação entre dois *primers* que amplificarão regiões genômicas distintas: um alvo (objeto de estudo), e um referência. Cada comparação será feita em dois tecidos diferentes: um conalteração e outro sem alteração. O primeiro passo da análise consiste em determinar a eficiência de cada par de *primer* utilizado na qPCR: um par no qual será feito o teste de deleção ou duplicação de um segmento, o *primer* alvo, e outro como referência em um segmento genômico normal, o *primer* referência. Essa eficiência é gerada a partir da seguinte equação:

```
        E= 10^((-1)///slope//), onde
```

E = eficiência; *slope* = coeficiente angular da reta de uma regressão linear.

O *slope* é obtido a partir de uma regressão linear entre o log10 da concentração da amostra e o respectivo valor de Ct (*cicle threshold*) de pelo menos cinco pontos de uma diluição seriada em triplicata de um gDNA (DNA genômico) ou cDNA, normalmente feitas em concentrações de: 25ng/µL, 12,5ng/µL, 6,25ng/µL, 3,125ng/µL e 1,5625ng/µL. Desse modo, teremos dois vetores de dados contendo a média da triplicata da Ct e o log10 da concentração correspondente. Para ser considerado eficiente, um par de *primer* deve apresentar valor de eficiência entre 95% e 105%, R2 ≥ 0,99, *slope* próximo a -3,32 e *intercept* maior que 32. Um valor de E igual a 2 significa que a cada ciclo da qPCR, a quantidade de DNA do segmento amplificado pelo par de *primer* dobrará. Os valores da eficiência serão utilizados para se determinar a razão da amplificação dos dois pares de *primers* de um gDNA ou cDNA alvo e um controle:

```
        RQ = (Ea^(ΔCt_alvo))/(Er^(ΔCt_referência)), onde:
```

Ea = eficiência do *primer* alvo; Er = eficiência do *primer* referência; ΔCt_alvo = Ct(gDNA controle) - Ct(gDNA teste), para o *primer* alvo; ΔCt_referência = Ct(gDNA controle) - Ct(gDNA teste), para o *primer* de referência.

Um valor de (i) RQ ≈ 1 significa que não há diferença entre o segmento alvo e a referência; (ii) RQ ≈ 1,5 significa que o segmento amplificado pelo *primer* alvo está com uma cópia a mais no genoma em relação à referência; RQ ≈ 0,5 significa que o segmento amplificado pelo *primer* alvo está com uma cópia a menos no genoma em relação à referência. Valores intermediários a 1,5 ~ 1 e 1 ~ 0,5, indicam que a deleção está em mosaico, a deleção não é germinativa e sim somática.

As análises de qPCR permitem três tipos de abordagens:

1°) **Não se considera a eficiência efetiva** de cada *primer*, sendo assim a RQ obtida a partir de:

```
        RQ = (2^(ΔCt_alvo))/(2^(ΔCt_referência)),
```

onde cada *primer* tem sua eficiência considerada perfeita, ou seja, 2;

2°) Se considera a eficiência efetiva de cada *primer* **independente dos desvios (sd) de cada triplicata**; e

3°) Se considera a eficiência efetiva de cada *primer* **com desvios (sd) de cada triplicata inferior a 0,2**. Para isso, se remove uma das triplicatas de modo a reduzir o desvio ao menor possível e a regressão linear de cada *primer* obtida através de uma media de duas Ct.

Desse modo o usuário da função poderá optar por três tipos de análise: (i) sem considerar a eficiência efetiva de cada par de *primer*, (ii) considerando-se a eficiência e desconsiderando os desvios, e (iii) considerando-se a eficiência e os desvios.

Para a análise i, o *output* será a eficiência de 2, para cada um dos dois *primer*, o alvo e o referência, além da RQ. Para as análises ii e iii teremos 5 *outputs* diferentes: um plot do modelo linear para o *primer* referência e outro para o alvo, um plot de análise geral dos dados do modelo linear contendo os dados de Residual vs Fitted, Normal Q-Q, Scale-Location e Residual vs Leverage, para o *primer* referência e outro para o alvo e uma tabela final contendo os valores do *Intercept*, *Slope*, Eficiência e RQ. Todos os dados gráficos serão guardados automaticamente no diretório de trabalho do usuário.

No caso de optar pela análise 2, o usuário será alertado com uma mensagem de warning sobre possíveis sd superiores a 0,2, possibilitando que ele mude para a análise 3. Para o usuário, o *input* de dados será simples: um arquivo em cvs. contendo 8 colunas simples para Well, Sample_Name, DNA_Concentration, Tag, Target_Name, Reporter, Ct, Ct_Threshold.

Well	Sample_Name	DNA_Concentration	Tag	Target_Name	Reporter	Ct	Ct_Threshold
A1	25ng\ul	25	Efficiency	GADPH	SYBR	22,28717232	0,433986658
A2	25ng\ul	25	Efficiency	GADPH	SYBR	22,35695076	0,433986658
A3	25ng\ul	25	Efficiency	GADPH	SYBR	22,45048714	0,433986658

Desse modo, com uma tabela simples contendo as informações de qual o nome das amostras (Sample_Name), as concentrações (DNA_Concentration), se será usado para teste de eficiência (Tag) e o nome do *primer* (Target_Name) a aplicação de uma função à análise estatística da metodologia ΔΔCt para a análise de expressão ou de duplicação\deleção genômica poupará tempo, visto que as análises são recorrentes nesse procedimento.

Help

```
package: qpcr                R Documentation

Função definida para análise  de dados obtidos por PCR em tempo real pela
metodologia ΔΔCt.

Description:

A técnica de qPCR (quantitative polymarase chain reaction) é largamente
utilizada na quantificação da expressão gênica ou na comparação de segmentos
genômicos deletados ou duplicados. A análise de qPCR pela metodologia ΔΔCt
consiste na comparação entre duas regiões genômicas distintas ou RNAs
distintos: um alvo (objeto de estudo), e um referência (controle). A função
qpcr permite a partir de um input de dados no formato .csv, as análises sem
levar em consideração a eficiência dos primers, ou levando-se em
```

consideração as eficiência dos primers, ou os menores desvios de cada triplicata.

Usage:

qpcr = function(qpcr.data, ref, trg, efi = TRUE, adj = FALSE)

Arguments:

  * qpcr.data = Dados no formato .csv contendo 8 colunas simples para Well, Sample_Name, DNA_Concentration, Tag, Target_Name, Reporter, Ct, Ct_Threshold. Well permite dados numéricos e não numéricos referentes a posição da amostar numa placa ou tubos; Sample_Name permite dados numéricos e não numéricos referentes ao nome da amostra; DNA_Concentration permite somente dados numéricos sendo as casas decimais separadas por . ; Tag permite somente as identificações da amostra como sendo Efficiency (para cálculo de eficiência dos primers) e Test (para cálculo de ΔΔCt e RQ), Target_Name permite dados numéricos e não numéricos referentes ao nome do primer; Reporter se a amostra contém Syber Green ou TaqMan; Ct permite somente dados numéricos sendo as casas decimais separadas por . ; Ct_Threshold permite somente dados numéricos sendo as casas decimais separadas por ..
  * ref = Nome do primer referência assim como descrito em qpcr.data.
  * trg = Nome do primer alvo assim como descrito em qpcr.data.
  * efi = Por default efi = TRUE. Permite o usuário escolher a análise de ΔΔCt utilizando as eficiências dos primers do próprio experimento. Se efi = FALSE, as eficiências de cada primer será considerada 2.
  * adj = Por default adj = FALSE. A análise de ΔΔCt utilizará a média dos valores das triplicatas experimentais sem levar em conta a sd. Se adj = TRUE, a análise utilizará a menor sd de duas amostras das triplicatas experimentais.

Details:

A função qpcr tem por objetivo facilitar a análise dos dados de qPCR feitos a partir de triplicatas de todas as amostras, não necessitando porem que seja feito no mesmo experimento o teste de eficiência dos primers. Nesse último caso, a eficiência de cada primer será considerada perfeita, ou seja, 2. Os dados contidos no arquivo qpcr.data deve conter exatamente a ordem das colunas acima determinado.
Tanto ref quanto trg devem estar escritos como consta em qpcr.data. Do mesmo modo, a coluna Tag deve conter as informações Efficiency ou Test, extremamente necessário para o cálculo da ΔΔCt e eficiências dos primers. Em ambos os casos a análise pode parar ou mesmo retornar avisos de warning (veja detalhes em Warning).
Os gráficos das análises (veja detalhes em Value) são salvos automaticamente na pasta de trabalho do usuário toda vez em que a análise for rodada.

Value:

```
No máximo cinco outputs são processados automaticamente na análise default
da função qpcr: plot_lm_ref, plot_lm_trg, plot_mod_ref, plot_mod_trg e
resultado contendo em forma de tabela o intercept, slope, eficiência de cada
primer, além de Rq.
Quando a função ajustada para adj = TRUE, outros cinco outputs são
processados automaticamente: plot_lm_adj_ref, plot_lm_adj_trg,
plot_mod_adj_ref, plot_mod_adj_trg e resultado contendo em forma de tabela o
intercept, slope, eficiência de cada primer, além de Rq.
A função setada para efi = TRUE, somente o valor de Rq será gerado.
Todos os gráficos são salvos automaticamente na pasta de trabalho do
usuário. Estes arquivos são gerados no formato .tif.

  * plot_lm_ref e plot_lm_trg: Gráfico com a regressão linear da média das
triplicatas de Ct tanto para referência quanto para alvo. As legendas
apresentam o valor do intercept e slope da reta.

  * plot_mod_ref e plot_mod_trg: Quatro gráficos de análise geral dos dados
do modelo linear contendo os dados de Residual vs Fitted, Normal Q-Q, Scale-
Location e Residual vs Leverage, tanto para referência e quanto para alvo.

  * plot_lm_adj_ref e plot_lm_adj_trg: Gráfico com a regressão linear da
média ajustada a partir das duas amostras de Ct que resultam no menor sd
tanto para referência quanto para alvo. As legendas apresentam o valor do
intercept e slope da reta ajustada.

  * plot_mod_adj_ref e plot_mod_adj_trg: Quatro gráficos de análise geral
dos dados do modelo linear ajustado a menor sd contendo os dados de Residual
vs Fitted, Normal Q-Q, Scale-Location e Residual vs Leverage, tanto para
referência e quanto para alvo.

Warning:

A função não rodará caso não esteja especificado Efficiency ou Test na
coluna Tag.
Caso alguma célula na coluna Ct esteja vazia, retornará um warning avisando
de qual célula está vazia.

Author(s):

Luiz Gustavo Dufner de Almeida, PhD Student, Laboratório de Genética
Funcional - IB - USP.

References:

   *Bustin, S. A., et al. (2009). "The MIQE guidelines: minimum information
for publication of quantitative real-time PCR experiments." Clin Chem 55(4):
611-622.
   *Livak, K. J. and T. D. Schmittgen (2001). "Analysis of relative gene
expression data using real-time quantitative PCR and the 2(-Delta Delta
C(T)) Method." Methods 25(4): 402-408.
```

```
See Also:

   *Ginzinger, D. G. (2002). "Gene quantification using real-time
quantitative PCR: An emerging technology hits the mainstream." Experimental
hematology 30(6): 503-512.

Examples:

   *Criando dados para qpcr:
Sample_Name = LETTERS[rep(seq(from = 1, to = 16, by = 1), each = 3)]
DNA_Concentration = rep(rep(c(25, 12.5, 6.25, 3.125, 1.5625, 12.5, 12.5,
NA), each = 3), times = 2)
Tag = rep(rep(c("Efficiency", "Test"), times = c(15 , 9)), times = 2)
Target_Name = rep(c("primer_ref", "primer_trg"), each = 24)
Reporter = rep("Syber", times = 48)
Ct1 =  c(rnorm(3, mean = 21, sd = 0.12), rnorm(3, mean = 22, sd = 0.12),
rnorm(3, mean = 23, sd = 0.12), rnorm(3, mean = 24, sd = 0.12), rnorm(3,
mean = 25, sd = 0.12), rnorm(3, mean = 22.7, sd = 0.12), rnorm(3, mean =
22.5, sd = 0.12), NA, NA, NA)
Ct2 =  c(rnorm(3, mean = 22, sd = 0.12), rnorm(3, mean = 23, sd = 0.12),
rnorm(3, mean = 24, sd = 0.12), rnorm(3, mean = 25, sd = 0.12), rnorm(3,
mean = 26, sd = 0.12), rnorm(3, mean = 24.8, sd = 0.12), rnorm(3, mean =
23.7, sd = 0.12), NA, NA, NA)
Ct = c(Ct1, Ct2)
Ct_Threshold = rep(c(0.34567, 0.43219), each = 24)

   *Criando data-frame:
qpcr.ex = as.data.frame(cbind(Sample_Name, DNA_Concentration, Tag,
Target_Name, Reporter, Ct, Ct_Threshold), stringsAsFactors = FALSE)
qpcr.ex$DNA_Concentration = as.numeric(qpcr.ex$DNA_Concentration)
qpcr.ex$Ct = as.numeric(qpcr.ex$Ct)
qpcr.ex$Ct_Threshold = as.numeric(qpcr.ex$Ct_Threshold)

   *Salvando data-frame:
write.csv(qpcr.ex, file = "qpcr.ex.csv")

   *Rodando função qpcr:
qpcr("qpcr.ex.csv", "primer_ref", "primer_trg")
```

Função

dados_qpcr.csv

qpcr_function.r

```
qpcr = function(qpcr.data, ref, trg, efi = TRUE, adj = FALSE)
{
  qpcr = read.table(qpcr.data, header = TRUE, sep = ",", dec = ".", as.is =
TRUE)
  colnames(qpcr) = c("Well", "Sample_Name", "DNA_Concentration", "Tag",
```

```
"Target_Name", "Reporter", "Ct", "Ct_Threshold")
  for (i in seq(from = 1, to = length(qpcr$Ct)))
  {
    if (qpcr$Tag[i]!="Efficiency" & qpcr$Tag[i]!="Test")
    {
      stop(qpcr$Sample_Name[i], "This isn´t Efficiency or Test tags!")
    }
  }
  for (i in seq(from = 1, to = length(qpcr$Ct)))
  {
    if (is.na(qpcr$Ct[i]))
    {
      warning(qpcr$Sample_Name[i], "\n Exists NA in the column Ct! \n\t If
this is a negative control, disregard this warning!")
    }
  }
  ###################################
  ##Data frame of reference primer##
  ###################################
  ref.qpcr = data.frame(qpcr[qpcr$Target_Name==ref,])
  ref.qpcr = na.omit(ref.qpcr)
  ##Mean and sd of Ct##
  ref.qpcr.data = aggregate(ref.qpcr$Ct, by = list(ref.qpcr$Sample_Name,
ref.qpcr$DNA_Concentration, ref.qpcr$Tag), FUN = mean)
  ref.sd = aggregate(ref.qpcr$Ct, by = list(ref.qpcr$Sample_Name,
ref.qpcr$DNA_Concentration, ref.qpcr$Tag), FUN = sd)
  ref.qpcr.data$Sd_Ct = ref.sd[, 4]
  colnames(ref.qpcr.data) = c("Sample_Name", "DNA_Concentration","Tag",
"Mean_Ct", "Sd_Ct")
  ##Adjustable mean of minimum sd##
  ref.qpcr.dada.adj = as.data.frame(matrix(NA, nrow =
length(ref.qpcr.data$Sd_Ct), ncol = 5))
  colnames(ref.qpcr.dada.adj) = c("Sample_Name", "DNA_Concentration", "Tag",
"Mean_Ct_adj", "Sd_Ct_adj")
  j = 0
  for (i in seq(from = 1, to = length(ref.qpcr$Ct), by = 3))
  {
    j = j+1
    efi.r.sd1  = sd(ref.qpcr$Ct[c(i, i+1)])
    efi.r.sd2  = sd(ref.qpcr$Ct[c(i+1, i+2)])
    efi.r.sd3  = sd(ref.qpcr$Ct[c(i, i+2)])
    ref.qpcr.dada.adj[j, 5] = min(efi.r.sd1, efi.r.sd2, efi.r.sd3)
    if(efi.r.sd1 < efi.r.sd2 & efi.r.sd3)
    {
      mn  = mean(ref.qpcr$Ct[c(i, i+1)])
      ref.qpcr.dada.adj[j, 4] = mn
    }
    if(efi.r.sd2 < efi.r.sd1 & efi.r.sd3)
    {
      mn = mean(ref.qpcr$Ct[c(i+1, i+2)])
      ref.qpcr.dada.adj[j, 4] = mn
```

```
    }
    if(efi.r.sd3 < efi.r.sd1 & efi.r.sd2)
    {
      mn = mean(ref.qpcr$Ct[c(i, i+2)])
      ref.qpcr.dada.adj[j, 4] = mn
    }
    ref.qpcr.dada.adj[j, 1] = ref.qpcr[i, 2]
    ref.qpcr.dada.adj[j, 2] = ref.qpcr[i, 3]
    ref.qpcr.dada.adj[j, 3] = ref.qpcr[i, 4]
  }
  ################################
  ##Data frame of target primer##
  ################################
  trg.qpcr = data.frame(qpcr[qpcr$Target_Name==trg,])
  trg.qpcr = na.omit(trg.qpcr)
  ##Mean and sd of Ct##
  trg.qpcr.data = aggregate(trg.qpcr$Ct, by = list(trg.qpcr$Sample_Name,
trg.qpcr$DNA_Concentration, trg.qpcr$Tag), FUN = mean)
  trg.sd = aggregate(trg.qpcr$Ct, by = list(trg.qpcr$Sample_Name,
trg.qpcr$DNA_Concentration, trg.qpcr$Tag), FUN = sd)
  trg.qpcr.data$Sd_Ct = trg.sd[, 4]
  colnames(trg.qpcr.data) = c("Sample_Name", "DNA_Concentration", "Tag",
"Mean_Ct", "Sd_Ct")
  ##Adjustable mean of minimum sd##
  trg.qpcr.dada.adj = as.data.frame(matrix(NA, nrow =
length(trg.qpcr.data$Sd_Ct), ncol = 5))
  colnames(trg.qpcr.dada.adj) = c("Sample_Name", "DNA_Concentration", "Tag",
"Mean_Ct_adj", "Sd_Ct_adj")
  j = 0
  for (i in seq(from = 1, to = length(trg.qpcr$Ct), by = 3))
  {
    j = j+1
    efi.t.sd1  = sd(trg.qpcr$Ct[c(i, i+1)])
    efi.t.sd2  = sd(trg.qpcr$Ct[c(i+1, i+2)])
    efi.t.sd3  = sd(trg.qpcr$Ct[c(i, i+2)])
    trg.qpcr.dada.adj[j, 5] = min(efi.t.sd1, efi.t.sd2, efi.t.sd3)
    if(efi.t.sd1 < efi.t.sd2 & efi.t.sd3)
    {
      mn  = mean(trg.qpcr$Ct[c(i, i+1)])
      trg.qpcr.dada.adj[j, 4] = mn
    }
    if(efi.t.sd2 < efi.t.sd1 & efi.t.sd3)
    {
      mn = mean(trg.qpcr$Ct[c(i+1, i+2)])
      trg.qpcr.dada.adj[j, 4] = mn
    }
    if(efi.t.sd3 < efi.t.sd1 & efi.t.sd2)
    {
      mn = mean(trg.qpcr$Ct[c(i, i+2)])
      trg.qpcr.dada.adj[j, 4] = mn
```

```
    }
    trg.qpcr.dada.adj[j, 1] = trg.qpcr[i, 2]
    trg.qpcr.dada.adj[j, 2] = trg.qpcr[i, 3]
    trg.qpcr.dada.adj[j, 3] = trg.qpcr[i, 4]
  }
  if(efi==TRUE)
  {
    ############################
    ##Calculating efficiency##
    ############################
    if(adj==FALSE)
    {
      ##Linear regression of reference primer##
      ref.mod =
lm(ref.qpcr.data$Mean_Ct~log10(ref.qpcr.data$DNA_Concentration), subset =
ref.qpcr.data$Tag=="Efficiency")
      coef.ref = as.numeric(coef(ref.mod))
      e.ref = 10^(-1/coef.ref[2])
      x11()
      par(mfrow = c(2, 2))
      plot(ref.mod, main = ref)
      dev.copy(tiff, filename = "plot_mod_ref.tif");
      dev.off ()
      par(mfrow = c(1, 1))
      x11()
      plot(ref.qpcr.data$Mean_Ct~log10(ref.qpcr.data$DNA_Concentration),
subset = ref.qpcr.data$Tag=="Efficiency",
           xlab = "",
           ylab = "",
           main = ref, bty = "l", tcl = 0.3, col = "Blue", pch = 16)
      abline(ref.mod, lty = 1, col = "orange")
      mtext("DNA Log10 concentration", side = 1, line = 3, cex = 1)
      mtext("Mean Ct", side = 2, line = 3, cex = 1)
      legend("bottomleft", legend = c("Mean Ct", "Linear model of reference
primer", coef.ref), cex = 0.8, col = c("blue", "orange", "orange",
"orange"), pch = c(16, 32, 32, 32), lty = c(0, 1, 0, 0), bty = "n")
      dev.copy(tiff, filename = "plot_lm_ref.tif");
      dev.off ()
      ##Linear regression of target primer##
      trg.mod =
lm(trg.qpcr.data$Mean_Ct~log10(trg.qpcr.data$DNA_Concentration), subset =
trg.qpcr.data$Tag=="Efficiency")
      coef.trg = as.numeric(coef(trg.mod))
      e.trg = 10^(-1/coef.trg[2])
      x11()
      par(mfrow = c(2, 2))
      plot(trg.mod, main = trg)
      dev.copy(tiff, filename = "plot_mod_trg.tif");
      dev.off ()
      par(mfrow = c(1, 1))
      x11()
```

```
      plot(trg.qpcr.data$Mean_Ct~log10(trg.qpcr.data$DNA_Concentration),
subset = trg.qpcr.data$Tag=="Efficiency",
           xlab = "",
           ylab = "",
           main = trg, bty = "l", tcl = 0.3, col = "red", pch = 16)
      abline(trg.mod, lty = 1, col = "orange")
      mtext("DNA Log10 concentration", side = 1, line = 3, cex = 1)
      mtext("Mean Ct", side = 2, line = 3, cex = 1)
      legend("bottomleft", legend = c("Mean Ct", "Linear model of reference
primer", coef.trg), cex = 0.8, col = c("red", "orange", "orange", "orange"),
pch = c(16, 32, 32, 32), lty = c(0, 1, 0, 0), bty = "n")
      dev.copy(tiff, filename = "plot_lm_trg.tif");
      dev.off ()
      ######################
      ##delta-delta Ct Rq##
      ######################
      test.ref = data.frame(ref.qpcr.data[ref.qpcr.data$Tag=="Test",])
      test.ref = test.ref[order(test.ref$Sample_Name), ]
      test.trg = data.frame(trg.qpcr.data[trg.qpcr.data$Tag=="Test",])
      test.trg = test.trg[order(test.trg$Sample_Name), ]
      results = as.data.frame(matrix(NA, nrow = 3, ncol =
length(test.ref$Mean_Ct)))
      for (i in seq(from = 1, to = length(test.ref$Mean_Ct)))
      {
        colnames(results)[i] = test.ref$Sample_Name[i]
        rownames(results) = c("RQ", "Efficiency reference", "Efficiency
target")
      }
      for (i in seq(from = 1, to = length(test.ref$Mean_Ct)))
      {
        dif.ref = test.ref$Mean_Ct[i] - test.ref$Mean_Ct[1]
        dif.trg = test.trg$Mean_Ct[i] - test.trg$Mean_Ct[1]
        rq = (e.trg^dif.trg)/(e.ref^dif.ref)
        results[1, i] = rq
        results[2, i] = e.trg
        results[3, i] = e.ref
      }
      return(results)
    }
    else
    {
      ##Linear regression of reference primer##
      ref.mod.adj = lm(ref.qpcr.dada.adj$Mean_Ct_adj ~
log10(ref.qpcr.dada.adj$DNA_Concentration), subset =
ref.qpcr.dada.adj$Tag=="Efficiency")
      coef.ref.adj = as.numeric(coef(ref.mod.adj))
      e.ref.adj = 10^(-1/coef.ref.adj[2])
      x11()
      par(mfrow = c(2, 2))
      plot(ref.mod.adj, main = ref)
```

```
      dev.copy(tiff, filename = "plot_mod_adj_ref.tif");
      dev.off ()
      par(mfrow = c(1, 1))
      x11()
      plot(ref.qpcr.dada.adj$Mean_Ct_adj ~
log10(ref.qpcr.dada.adj$DNA_Concentration), subset =
ref.qpcr.dada.adj$Tag=="Efficiency",
           xlab = "",
           ylab = "",
           main = ref, bty = "l", tcl = 0.3, col = "Blue", pch = 16)
      abline(ref.mod.adj, lty = 1, col = "orange")
      mtext("DNA Log10 concentration", side = 1, line = 3, cex = 1)
      mtext("Mean Ct", side = 2, line = 3, cex = 1)
      legend("bottomleft", legend = c("Mean Ct", "Linear model of reference
primer", coef.ref.adj), cex = 0.8, col = c("blue", "orange", "orange",
"orange"), pch = c(16, 32, 32, 32), lty = c(0, 1, 0, 0), bty = "n")
      dev.copy(tiff, filename = "plot_lm_adj_ref.tif");
      dev.off ()
      ##Linear regression of target primer##
      trg.mod.adj =
lm(trg.qpcr.dada.adj$Mean_Ct_adj~log10(trg.qpcr.dada.adj$DNA_Concentration),
subset = trg.qpcr.dada.adj$Tag=="Efficiency")
      coef.trg.adj = as.numeric(coef(trg.mod.adj))
      e.trg.adj = 10^(-1/coef.trg.adj[2])
      x11()
      par(mfrow = c(2, 2))
      plot(trg.mod.adj, main = trg)
      dev.copy(tiff, filename = "plot_mod_adj_trg.tif");
      dev.off ()
      par(mfrow = c(1, 1))
      x11()
plot(trg.qpcr.dada.adj$Mean_Ct_adj~log10(trg.qpcr.dada.adj$DNA_Concentration
), subset = trg.qpcr.dada.adj$Tag=="Efficiency",
           xlab = "",
           ylab = "",
           main = trg, bty = "l", tcl = 0.3, col = "red", pch = 16)
      abline(trg.mod.adj, lty = 1, col = "orange")
      mtext("DNA Log10 concentration", side = 1, line = 3, cex = 1)
      mtext("Mean Ct", side = 2, line = 3, cex = 1)
      legend("bottomleft", legend = c("Mean Ct", "Linear model of reference
primer", coef.trg.adj), cex = 0.8, col = c("red", "orange", "orange",
"orange"), pch = c(16, 32, 32, 32), lty = c(0, 1, 0, 0), bty = "n")
      dev.copy(tiff, filename = "plot_ml_adj_trg.tif");
      dev.off ()
      ######################
      ##delta-delta Ct Rq##
      ######################
      test.ref.adj =
data.frame(ref.qpcr.dada.adj[ref.qpcr.dada.adj$Tag=="Test",])
      test.ref.adj = test.ref.adj[order(test.ref.adj$Sample_Name), ]
      test.trg.adj =
```

```
data.frame(trg.qpcr.dada.adj[trg.qpcr.dada.adj$Tag=="Test",])
      test.trg.adj = test.trg.adj[order(test.trg.adj$Sample_Name), ]
      results.adj = as.data.frame(matrix(NA, nrow = 3, ncol =
length(test.ref.adj$Mean_Ct)))
      for (i in seq(from = 1, to = length(test.ref.adj$Mean_Ct)))
      {
        colnames(results.adj)[i] = test.ref.adj$Sample_Name[i]
        rownames(results.adj) = c("RQ", "Efficiency reference", "Efficiency
terget")
      }
      for (i in seq(from = 1, to = length(test.ref.adj$Mean_Ct)))
      {
        dif.ref.adj = test.ref.adj$Mean_Ct[i] - test.ref.adj$Mean_Ct[1]
        dif.trg.adj = test.trg.adj$Mean_Ct[i] - test.trg.adj$Mean_Ct[1]
        rq = (e.trg.adj^dif.trg.adj)/(e.ref.adj^dif.ref.adj)
        results.adj[1, i] = rq
        results.adj[2, i] = e.trg.adj
        results.adj[3, i] = e.ref.adj
      }
      return(results.adj)
    }
  }
  else
  {
    ######################
    ##delta-delta Ct Rq##
    ######################
    test.ref.2 = data.frame(ref.qpcr.data[ref.qpcr.data$Tag=="Test",])
    test.ref.2 = test.ref.2[order(test.ref.2$Sample_Name), ]
    test.trg.2 = data.frame(trg.qpcr.data[trg.qpcr.data$Tag=="Test",])
    test.trg.2 = test.trg.2[order(test.trg.2$Sample_Name), ]
    results.2 = as.data.frame(matrix(NA, nrow = 3, ncol =
length(test.ref.2$Mean_Ct)))
    for (i in seq(from = 1, to = length(test.ref.2$Mean_Ct)))
    {
      colnames(results.2)[i] = test.trg.2$Sample_Name[i]
      rownames(results.2) = c("RQ", "Efficiency reference", "Eficiency
target" )
    }
    for (i in seq(from = 1, to = length(test.ref.2$Mean_Ct)))
    {
      dif.ref.2 = test.ref.2$Mean_Ct[i] - test.ref.2$Mean_Ct[1]
      dif.trg.2 = test.trg.2$Mean_Ct[i] - test.trg.2$Mean_Ct[1]
      rq = (2^dif.trg.2)/(2^dif.ref.2)
      results.2[1, i] = rq
      results.2[2, i] = 2.00
      results.2[3, i] = 2.00
    }
    return(results.2)
  }
```

```
}
```

Proposta 2

Comparação entre métodos de quantificação de DNA/RNA

As qualidades e as reprodutividades de cada experimento em biologia molecular passa obrigatoriamente por uma boa qualidade na extração de DNA/RNA, seja de tecido animal ou vegetal. Dependendo dos fins, existem diversas metodologias para extração de DNA/RNA, sendo que todas, obrigatoriamente serão quantificada por um espectrofotometro.

Os espectrofotometro, apresentam por princípio a incidência de um feixe luminoso com um comprimento de onda pré-determinado sobre uma amostra de DNA ou RNA. A capitação do feixe luminoso que retorna, possibilita a quantificação do DNA ou RNA, assim como a sua qualidade, que é obtida pelas razões 260/280 e 260/230.

Sample ID	260 Raw	280 Raw	230 Raw	320 Raw	260	280	230	260/280	260/230	ng/µL
amostra 1	0,140	0,090	0,230	0,050	0,090	0,050	0,180	2,090	0,520	93,450
amostra 2	0,140	0,090	0,250	0,040	0,100	0,050	0,200	2,080	0,500	100,700
amostra 3	0,150	0,100	0,250	0,050	0,100	0,050	0,200	2,050	0,510	103,130
amostra 4	0,110	0,070	0,230	0,040	0,060	0,030	0,180	2,270	0,350	63,960

Os aparelhos mais comumente utilizados para a quantificação tanto de DNA quanto de RNA são o NanoDrop e Epoch. Como é de se esperar a quantificação em cada aparelho diverge um do outro, o que portanto, poderá influenciar na escolhas das amostras para os experimentos. Desse modo, a partir de um banco de dados contendo amostras que foram quantificadas em ambos os aparelhos será possível traçar uma correlação tanto entre as concentrações, 260/280 e 260/230 para ambos os aparelhos.

Utilizando-se de um teste t-Student pareado para concentração, 260/280 e 260/230 será obtido um valor de p no qual informará se os resultados de um aparelho para outro divergem ou não. Os *output* para essa análise serão uma regressão entre concentração, 260/280 e 260/230, além de informar nos próprios gráficos se existem *outliers* e uma re-análise sem esses dados para um novo teste t-Student.

Desse modo, a fidedignidade para a concentração de cada amostra, assim como para as qualidades serão aumentadas, diminuindo o risco de escolha de uma amostra inadequada para futuros experimentos.

> Olá Luiz Gustavo, a sua proposta 1 me parece uma boa. No entanto ainda simples. Tente pensar por exemplo em como fazer a mesma tarefa para diferentes primers ao mesmo tempo. Ainda, não ficou claro qual seria o output da função. Os valores de eficiência e razão de amplificação? É possível criar alguma saída gráfica usando diferentes cores para os valores de RQ nas classes 1, 1.5 e 0.5? Teste sua função com diferentes valores alvo e referência para se certificar que você implementou de maneira bem geral, cobrindo todas as opções.

Sobre a proposta 2, é **essencial** que você tenha um plano B. Por favor, poste aqui a sua proposta 2. Não há razão alguma, nem mesmo uma proposta 1 muito boa, que nos faça aceitar apenas uma proposta.

—-*Sara Mortara*

Ola Sara M.

Obrigado pelas dicas. As alterações e modificações das propostas foram feitas como pedido.

Abraços

—-*Luiz Gustavo*

Legal, Luiz. Não esqueça de além de incluir a função e o help no próprio wiki você também deve fazer o upload dos arquivos. Tome cuidado quando postar no formato code e usar a formatação do wiki, veja que não funciona no modo code. Melhor tirar a formatação de lá para facilitar a leitura. Veja exemplos de alunos dos anos anteriores. —-*Sara Mortara*

From:
http://ecor.ib.usp.br/ - **ecoR**

Permanent link:
http://ecor.ib.usp.br/doku.php?id=05_curso_antigo:r2016:alunos:trabalho_final:luiz.gustavo.almeida:start

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