ddCt method for qRT–PCR data analysis

Jitao David Zhang, Markus Ruschhaupt and Rudolf Biczok

October 17, 2016

Abstract

Here we describe the $2^{-\Delta\Delta C_T}$ algorithm implemented in the *ddCt* package. The package is designed for the data analysis of quantitative real–time PCR (qRT–PCR) experiemtns in Bioconductor. With the *ddCt* package, one can acquire the relative expression of the target gene in different samples. This vignette mainly dicusses the principles of the ddCt algorithm and demonstrates the functionality of the package with a compact example. Another vignette in the package, *rtPCR-usage*, gives instructions to call the script for end–to–end analysis.

Both absolute and relative quantification have been used to analyse the data from the quantitative real–time PCR (qRT–PCR, or RT–PCR for short) experiments. The $2^{-\Delta\Delta C_T}$ algorithm, also known as the the *delta-delta-Ct* or *ddCt* algorithm, is a convenient method to analyze the relative changes in gene expression **?**. It requires the assignment of one or more housekeeping genes, which are assumed to be uniformly and constantly expressed in all samples, as well as one or more reference samples. The expression of other samples is then compared to that in the reference sample[1].

1 RT–PCR

Rich background knowledge about RT–PCR can be found at **?**, **?** and **?**. There are many variations in the experimental processes. Here we shortly summarize the general key steps in the TaqMan®assay to help the understanding of following discussions.

1. RNA preparation: total or specific type of RNA are extracted from cell lines, tissues, biopsies, etc.

2. RNA is Reversed Transcribed into DNA, which is also known as the *RT-reaction*.

[1]The *qpcrNorm* package in the Bioconductor repository introduces the data–driven normalization method for high–throughput qPCR data, which does not depend on the house–keeping genes but makes extra assumptions. See the help pages and the vignette of the *qpcrNorm* package for further information

3. qPCR probes (sometimes also known as 'primers') are added to the transcribed cDNA sample and the polymerase chain reaction takes place. This probe is an oligonucleotide with a reporter dye attached to the 5' end and a quencher dye attached to the 3' end. Till the time the probe is not hydrolized, the quencher and the fluorophore remain in proximity to each other, which does not completely quench the flourescence of the reporter dye and therefore only a background flourescence is observed.

4. During PCR, the probe anneals specifically between the forward and reverse primer to an internal region of the PCR product. The polymerase then carries out the extension of the primer and replicates the template to which the TaqMan®is bound. The 5' exonuclease activity of the polymerase cleaves the probe, releasing the reporter molecule away from the close vicinity of the quencher. The fluorescence intensity of the reporter dye, as a result increases. This process repeats in every cycle.

5. As the cycle number increases, the detected fluorescence also increases. And when the fluorescence crosses an arbitrary line, the device recodes the cycle number until then, which is known as the C_T value.

In princple one could also report the C_T values of the housekeeping gene and the sample gene(s) in the form of barplots to show their relative relation. However, this has two main drawbacks:

- This is only applicable in cases where more than one genes are compared in the same sample. In case of mutilple samples one has to calculate the relative expression to a specified *reference sample*

- C_T value is exponential. In case of a ideal amplification efficiency of 1, increase of the C_T value by 1 indicates a two–fold expression. Therefore, it maybe misleading to illustrate the expression with the raw C_T value.

2 The *ddCt* Algorithm

The *ddCt* method was one of the first methods used to to calculate real–time PCR results. Different the standard curve **?** and the Pfaffl method **?**, *ddCt* is an approximation method and makes various assumptions. However, it reduces lot of experiment effort by making these assumptions and is easy to implement, and in many cases they return results similarly to other non-approximation methods **?**.

2.1 Deviation

The exponential amplification of the polymerase chain reaction (PCR) can be described by the equation **??**.

$$X_n = X_0 \times (1 + E_X)^n \tag{1}$$

where X_n is the number of target molecules at cycle n of the reaction, and X_0 is the number of target molecules initially. E_x is the amplification efficiency of target amplification, and n is the number of cycles. The threshold value (C_T) records the fractional cycle number at which the fluorescence reaches a fixed threshold (see section **??**). Therefore

$$X_T = X_0 \times (1 + E_X)^{C_{T,X}} = K_X \tag{2}$$

where X_T is the threshold number of target molecules, $C_{T,X}$ is the readout C_T value, and K_X is a constant. Similarly we can express the equation **??** for the endogenous reference gene (house-keeping genes) as

$$R_T = R_0 \times (1 + E_R)^{C_{T,R}} = K_R \tag{3}$$

where R_T is the threshold number of the reference molecules, R_0 is the initial number of reference molecules. E_R is the efficiency of reference amplification, $C_{T,R}$ is the C_T readout for the reference, and K_R is a constant.

Combining equation **??** and **??** we get

$$\frac{X_T}{R_T} = \frac{X_0 \times (1 + E_X)^{C_{T,X}}}{R_0 \times (1 + E_R)^{C_{T,R}}} = \frac{K_X}{K_R} = K \tag{4}$$

For qRT–PCR using TaqMan®probes, the exact values of X_T and R_T depend on several factors including the chemistry of reporter dye, the sequence context effects on the fluorescence properties of the probe, the fficiency of probe cleavage, purity of the probe, and the setting of the fluorescence threshold **?**. Therefor, the constant K does not have to be equal to one.

Assuming efficiencies of the target and the reference are the same,

$$\begin{aligned} E_X = E_R = E \\ \frac{X_0}{R_0} \times (1 + E)^{C_{T,X} - C_{T,R}} = K \end{aligned} \tag{5}$$

or

$$X_N \times (1 + E)^{\Delta C_T} = K \tag{6}$$

where X_N is equal to the *normalized* amount of target (X_0/R_0) and the dC_T is equal to the difference in the C_T for target and reference ($C_{T,X} - C_{T,R}$).

Equation **??** can be rearranged as

$$X_N = K \times (1 + E)^{-\Delta C_T} \tag{7}$$

The final step is to divide the X_N in the equation **??** for any sample q by the reference sample (also known as the calibrator, cb):

$$\frac{X_{N,q}}{X_{N,cb}} = \frac{K \times (1 + E)^{-\Delta C_{T,q}}}{K \times (1 + E)^{-\Delta C_{T,cb}}} = (1 + E)^{-\Delta\Delta C_T} \tag{8}$$

Here $-\Delta\Delta C_T = -(\Delta C_{T,q} - \Delta C_{T,cb})$.

For amplicons designed to be less than 150 basepairs and for which the primer and Mg^{2+} concentration have been optimized, the efficiency E is close to one. Therefore, the amount of target, normalized to the endogenous reference and relative to a reference sample, is given by

$$\text{amount of target} = 2^{-\Delta\Delta C_T} \tag{9}$$

Attention: Note that for the ddC_T calculation to be valid, the amplification efficiencies of the target and reference must be approximately equal.

3 Application example

Here we show how to use the *ddCt* package by a short example.

3.1 File I/O setup

We have attached two SDS output files, 'Experiment1.txt' and 'Experiment2.txt', in the package directory. The sample annotation information is also provided as the tab-delimited text file 'sampleData.txt'. Any warning information (for example *Undetermined* in reference sample) is saved as a text file specified by the parameter 'warningFile'.

```
> library(Biobase)
> library(lattice)
> library(RColorBrewer)
> library(ddCt)
> datadir <- function(x) system.file("extdata", x, package="ddCt")
> savedir <- function(x) file.path(tempdir(), x)
> file.names <- c(datadir("Experiment1.txt"),datadir("Experiment2.txt"))
> info <- datadir("sampleData.txt")
> warningFile <- savedir("warnings.txt")
```

3.2 Reference sample and housekeeping gene

For the sake of simplicity, we choose **Sample1** and **Sample2** as reference samples (calibrators), and **Gene2** as the reference gene (housekeeping gene) respectively. This could happen, for example, if the **Sample1** and **Sample2** are untreated samples while the **Sample3** has been treated with certain drugs. And **Gene2** is a housekeeping gene which we assume is expressed constantly in all the samples.

```
> name.reference.sample <- c("Sample1", "Sample2")
> name.reference.gene <- c("Gene2")
```

Note that more than one reference sample or renference gene can be specified.

3.3 Read in data

SDMFrame function is called to read in experiment data.Optionally one could also read in the sample annotation, which is the sampleInformation object in the example.

```
> library(Biobase)
> CtData <- SDMFrame(file.names)
> sampleInformation <- read.AnnotatedDataFrame(info,header=TRUE, row.names=NULL)
```

Note that SDMFrame is able to accept one or more files as input.

3.4 Apply the *ddCt* method

Next step we call ddCtExpression to perform *ddCt* method on the data.

```
> result <- ddCtExpression(CtData,
+                          calibrationSample=name.reference.sample,
+                          housekeepingGene=name.reference.gene,
+                          sampleInformation=sampleInformation,
+                          warningStream=warningFile)
```

Please refer to the help page of ddCtExpression-class for the methods to access all the values calculated by the *ddCt* method. For example the error (either standard deviation or median absolute deviation) of all replicates are accessible through

```
> CtErr(result)
```

```
        Sample
Detector    Sample1   Sample2    Sample3
   Gene1 0.72107613 0.1657458 0.45246203
   Gene2 0.06299529 0.0954406 0.03288257
   Gene3 0.44017062 0.6452858 0.17610413
        Sample
Detector    Sample4
   Gene1 0.05064879
   Gene2 0.03057597
   Gene3         NA
```

3.5 Visualization

errBarchart provides a simple way to visualize the experiment results with the modified barchart from the *lattice* package, as seen in the figure **??**

```
> br <- errBarchart(result)
> print(br)
```

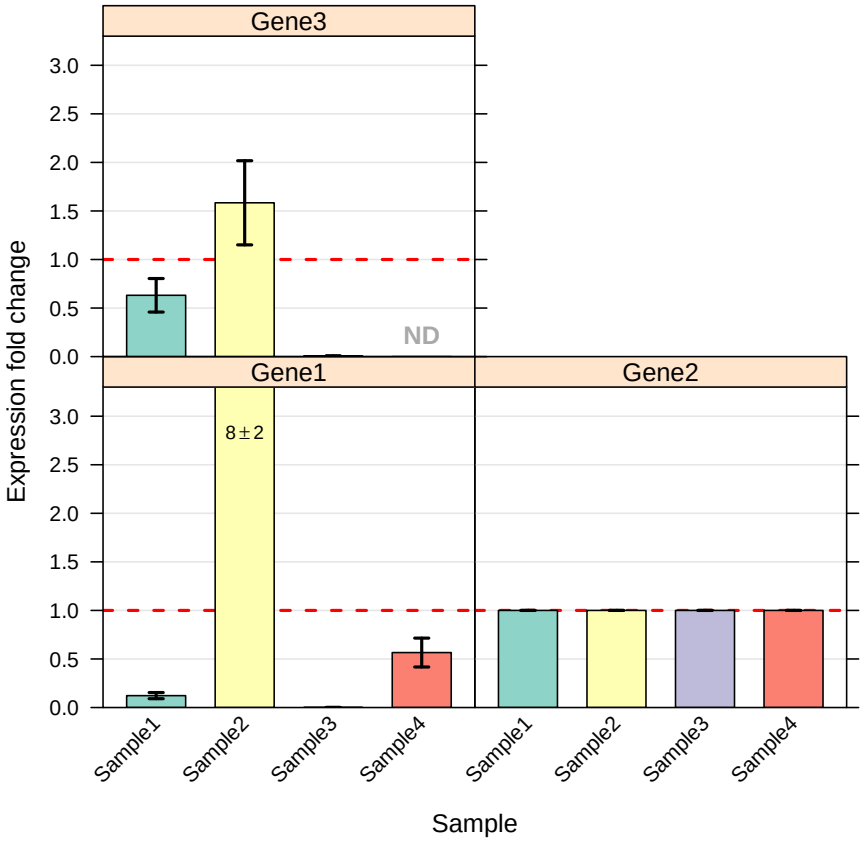

Figure 1: Barchart with error bars to visualize the expression fold change of target genes in samples. Each panel represents one target gene (Gene 1, 2 and 3 in this case), and the expression level in each sample is indicated by the height of bars. If one gene is not detected in one sample, a 'NA' symbol in grey will appear in the position of the bar (Gene3 in Sample4), which helps to differ from the situations where the expression is very low but not yet undetectable (for instance Gene3 in Sample3). See the help page of `errBarchart` for more details.

3.6 Write result to text file

Finally we can save the results as tab-delimited text files.

```
> elistWrite(result,file=savedir("allValues.txt"))
```

4 Acknowledgement

We thank Florian Hahne, Wolfgang Huber, Andreas Buness and Stefan Wiemann for their suggestion and comments during the development of the package. The example data has been kindly provided by Ute Ernst and was produced in the Division of Molecular Genome Analysis, DKFZ Heidelberg, Germany.

5 Session Info

The script has been running in the following session:

- R version 3.3.1 (2016-06-21), `x86_64-apple-darwin13.4.0`
- Base packages: base, datasets, grDevices, graphics, methods, parallel, stats, utils
- Other packages: Biobase 2.34.0, BiocGenerics 0.20.0, RColorBrewer 1.1-2, ddCt 1.30.0, lattice 0.20-34
- Loaded via a namespace (and not attached): grid 3.3.1, tools 3.3.1, xtable 1.8-2