HTqPCR

Daniela García Yuvia Pérez

> CCG UNAM

October 23, 2009

3

< □ > < □ > < □ > < □ > < □ >

Outline

- What is HTqPCR for?
- Reading in the raw data
- Data visualisation
- Feature categories and filtering
- Data Normalisation
- Filtering and subsetting the data
- Quality assessment
- Hierarchical clustering

< (T) >

Introduction I

The package HTqPCR is designed for the analysis of cycle threshold (Ct) values from quantitative real-time PCR data. The main areas of functionality comprise data import, quality assessment, normalisation, data visualisation, and testing for statistical significance in Ct values between different features (genes, miRNAs).

> library(HTqPCR)

< 日 > < 同 > < 三 > < 三 >

The package employs functions from other packages of the Bioconductor project. Dependencies include Biobase, RColorBrewer, limma, statmod, affy and gplots. Two qPCRset test objects are included in the package: one containing raw data, and the other containing processed values.

- > data(qPCRraw)
- > data(qPCRpros)

< ロ > < 同 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ >

General data format I

The input consists of tab-delimited text files containing the Ct values for a range of genes.

The package comes with example input files (from Applied Biosystem's TLDA cards), along with a text file listing sample file names and biological conditions.

- > path <- system.file("exData", package = "HTqPCR")</pre>
- > read.delim(file.path(path, "files.txt"))

File Treatment

- 1 sample1.txt Control
- 2 sample2.txt LongStarve
- 3 sample3.txt LongStarve
- 4 sample4.txt Control
- 5 sample5.txt Starve
- 6 sample6.txt Starve

< ロ > < 同 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ >

Overview of Ct values across all groups I

To get a general overview of the data the (average) Ct values for a set of features across all samples or different condition groups can be displayed.

< 日 > < 同 > < 三 > < 三 >



, 2009 8 / 34

Spatial layout I

When the features are organised in a particular spatial pattern, it is possible to plot the Ct values or other characteristics of the features using this layout.

< □ > < 同 > < 回 > < 回 > < 回 >

```
> plotCtCard(raw, col.range = c(10,
+ 35))
```

3

<ロト < 四ト < 三ト < 三ト

sample1



October 23, 2009 11 / 34

- > featureClass(raw) <- factor(c("Marker", + "TF", "Kinase")[sample(c(1, + 1 2 2 1 2) 284 replace = TPUE)])
- + 1, 2, 2, 1, 3), 384, replace = TRUE)])
- > plotCtCard(raw, plot = "class")

sample1



3

< □ > < □ > < □ > < □ > < □ >

Overview of Ct values I

Each Ct values in HTqPCR has an associated feature category. This is an important component to indicate the reliability of the qPCR data.

```
> raw.cat <- setCategory(raw, groups = files$Treatment,
+ quantile = 0.8)
```

Categories after Ct.max and Ct.min filtering: sample1 sample2 sample3 OK 313 264 327 Undetermined 68 119 56 Unreliable 3 1 1

> plotCtCategory(raw.cat)

3

<ロト < 四ト < 三ト < 三ト

Feature categories



October 23, 2009 16 / 34

æ

Normalisation I

Four different normalisation methods are currently implemented in HTqPCR. Two of these.

- quantile. Will make the distribution of Ct values more or less identical across samples.
- norm.rankinvariant. Computes all rank-invariant sets of features between pairwise comparisons of each sample against a reference, such as a pseudo-mean.
- scale.rankinvariant.Takes only the features found in a certain number of samples, and used the average Ct value of those as a scaling factor for correcting all Ct values.
- deltaCt. Calculates the standard deltaCt values, i.e. subtracts the mean of the chosen controls from all other values in the feature set.

Sources of biological sequences I

```
> q.norm <- normalizeCtData(raw.cat,</pre>
```

```
+ norm = "quantile")
```

```
> sr.norm <- normalizeCtData(raw.cat,
+ norm = "scale.rank")
```

```
Scaling Ct values
Using rank invariant genes: Gene1 Gene29
Scaling factors: 1.00 1.06 1.00 1.03 1.00 1.00
```

```
> nr.norm <- normalizeCtData(raw.cat,
+ norm = "norm.rank")
```

・ロト ・ 戸 ・ ・ ヨ ト ・ ヨ ・ うへつ

Sources of biological sequences II

```
Normalizing Ct values
Using rank invariant genes:
sample1: 75 rank invariant genes
```

```
> d.norm <- normalizeCtData(raw.cat,
+ norm = "deltaCt", deltaCt.genes = c("Gene1",
+ "Gene60"))
```

```
Calculating deltaCt values
Using control gene(s): Gene1 Gene60
Card 1: Mean=14.45 Stdev=4.25
```

Data Normalisation



Filtering and subsetting the data I

At any point during the analysis it's possible to filter out both individual features or groups of features that are either deemed to be of low quality, or not of interest for a particular aspect of the analysis. The qPCRset object can also be turned into smaller subsets, for example if only a particular class of features are to be used, or some samples should be excluded.

```
> nr.norm[1:10, ]
```

```
An object of class "qPCRset"
Size: 10 features, 6 samples
Feature types: Endogenous Control, Target
Feature names: Gene1 Gene2 Gene3 ...
Feature classes: Kinase, Marker, TF
Feature categories: OK, Unreliable, Undetermined
Sample names: sample1 sample2 sample3 ...
```

◆□▶ ◆□▶ ◆ □▶ ◆ □▶ ● □ ● ● ● ●

- > qFilt <- filterCtData(nr.norm,</pre>
- + remove.type = "Endogenous Control")

Removed 4 'Endogenous Control' features based on featureType(q)

```
> qFilt <- filterCtData(nr.norm,
+ remove.name = c("Gene1", "Gene20",
+ "Gene30"))
```

Removed 8 'Gene1/Gene20/Gene30' features based on featureNames

```
> qFilt <- filterCtData(nr.norm,</pre>
```

```
+ remove.class = "Kinase")
```

Removed 61 'Kinase' features based on featureClass(q).

▲□▶ ▲□▶ ▲□▶ ▲□▶ □ ● ● ●

```
> qFilt <- filterCtData(nr.norm,
+ remove.type = c("Endogenous Control"),
+ remove.name = c("Gene1", "Gene20",
+ "Gene30"))
```

Removed 4 'Endogenous Control' features based on featureType(q) Removed 4 'Gene1/Gene20/Gene30' features based on featureNames

E SQA

Quality assesment I

The overall correlation between different samples can be displayed visually.

Daniela GarcíaYuvia Pérez (CCG UNAM)

October 23, 2009 24 / 34

э

< □ > < 同 > < 回 > < 回 > < 回 >

> plotCtCor(raw, main = "Ct correlation")

- 2

Quality assesment



Ct correlation



Distribution of Ct values I

It may be of interest to examine the general distribution of data both before and after normalisation.

> summary(raw)

	sample1	sample2	<pre>sample3</pre>
Min.	" 7.218"	" 7.408"	" 6.19"
1st Qu.	"26.738"	"28.855"	"27.90"
Median	"28.937"	"30.994"	"29.92"
Mean	"29.542"	"32.190"	"30.35"
3rd Qu.	"33.323"	"35.985"	"32.98"
Max.	"40.000"	"40.000"	"40.00"

3

イロト イポト イヨト イヨト

- > par(mfrow = c(1, 2))
- > plotCtDensity(sr.norm)
- > plotCtHistogram(sr.norm)

イロト イポト イヨト イヨト



sample1

Daniela GarcíaYuvia Pérez (CCG UNAM)

October 23, 2009 29 / 34

Hierarchical clustering I

Both features and samples can be subjected to hierarchical clustering to display similarities and differences within groups of data. Individual subclusters can be selected.

< □ > < □ > < □ > < □ > < □ > < □ >

```
> c6 <- cluster.list[[6]]
> sr.norm[c6, ]
```

An object of class "qPCRset" Size: 11 features, 6 samples Feature types: Endogenous Control, Target Feature names: Gene9 Gene13 Gene46 ... Feature classes: Kinase, Marker, TF Feature categories: Undetermined, Unreliable, OK Sample names: sample1 sample2 sample3 ...

(日)

- > cluster.list <- clusterCt(sr.norm,</pre>
- + type = "genes", n.cluster = 6,
- + cex = 0.5)

▲□▶ ▲□▶ ▲□▶ ▲□▶ □ ● ● ●



Height

Daniela GarcíaYuvia Pérez (CCG UNAM)

October 23, 2009 33 / 34

æ

< □ > < □ > < □ > < □ > < □ >

Auf Wiedersehen :)

End

Au revoir :D

3

<ロト < 四ト < 三ト < 三ト