eman ta zabal zazu

Universidad del País Vasco Euskal Herriko Unibertsitatea

Bachelor in Informatics Engineering
Computation

Bachelor Dissertation

Quantitative real-time PCR data analysis with R

Author
Ignacio MACHADO

Supervisors
Borja CALVO
Iñaki INZA

informatika fakultatea facultad de informática

2016

Acknowledgements

In this section, I would like to thank everybody that show any kind of concern on my project, from the smallest to the biggest, as well as everybody that has been (even a little) part of my life influencing on me to actually make this project finally possible. In the next paragraphs, I am going to refer to some special people that I would like to specially thank.

First of all, my parents and my sister. Thank you for being with me in the moments of insecurity and uncertainty, in good times and bad, even though you got no clue about I was doing, you supported me cheering me on and you did all that you could have done. I cannot be less proud of you.

Second, my supervisors: Borja Calvo and Iñaki Inza. Two of the nicest professors I have ever met that supported me on everything related to the project, I have no doubts on how much they know and they have been always ready to help. Thank you for giving me the possibility to actually take a glance on this field. It would not be possible without you.

Third, my friends who have a big influence on me and my university colleagues. This project would not be possible if I could not spend time, laugh and relax with you. You have been my best source of disconnection from the project.

Finally, the researchers from Leioa, Maite Arzuaga and Itziar Irigoien. You all have been important to me. Special thanks to Andone Estomba and Maite for supporting me at the beginning of the project when I had no clue about how to start it. You helped me to understand the project from another way of thinking, which I strongly empathise with. Special thanks to Itziar for helping me with the understanding of statistics concepts and being ready to help me at any time she could.

UNIVERSITY OF THE BASQUE COUNTRY

Abstract

Faculty of Informatics

Bachelor in Informatics Engineering

Quantitative real-time PCR data analysis with R

by Ignacio MACHADO

This dissertation covers an introduction to the field of qPCR data analysis using the state-of-the-art R programming language. qPCR analyses genomic data based on the DNA replication. After showing a set of basic biological and statistics concepts, the roots of qPCR technology, together with its principal data analysis modelisations (visualisation, filtering, clustering, ...) are shown. A web application has been developed to ease and extend qPCR data analysis functionalities to other areas such as biology or forensics.

Contents

Chapter 1

Introduction

1.1 Project motivation

I feel very strongly that it was important that the final project work had a motivation behind for me. A project that would make my time worth and something that I would really enjoy from the beginning until the end. This project work has been motivated for several reasons that I'm explaining further on within the next paragraphs.

Although I was aware of my IT vocation since I was practically a child, my favourite subjects in the school were always related to science. I was particularly passionate about natural sciences when I was in primary school and I realised that Chemistry and Biology were my favourite subjects later on. I always showed a special curiosity in those and I specially enjoyed classes when we were doing experiments because I was curious about how theory was put into practice.

Moreover, one of my closest friend is, to this day, fighting against a chronic tumour and I had to support him in the last years. I believe this has influenced me on getting into the bioinformatics field and wondering what I can do to help.

When I was choosing my speciality for the third year, I was in doubts between software and computation, but my interest in doing a bioinformatics related project was determining. Anyway, I had the opportunity to develop a software application within my project scope.

1.2 Project definition

The aim of the project consists of having an introduction on the bioinformatics field as a bachelor final project work. This project is planned to have 12 ECTS credits inside the Bachelor Degree in Informatics Engineering and this means an average of 300 hours of work.
This introduction on the huge bioinformatics field has two different insights inside the project: on the one hand, Bioinformatics as a research field; on the other hand, Bioinformatics as a service. Two insights that are linked into each other within this project.

Bioinformatics as a research field Bioinformatics as a research field is materialised in the comprehension and analysis of qPCR expression data. Comprehension and analysis of this kind of data requires of flexible and reliable tools. Being R (Crawley, 2007) so frequently used in the bioinformatics field ignited our curiosity to research deeper into this programming language, as it gives us the necessary flexibility and reliability to analyse data. The state-of-the-art free, open source and open development project in the field of genomic data is called Bioconductor and it provides more than a thousand packages in R. Among the huge amount of packages, there are packages to analyse specifically qPCR data such as HTqPCR, SLqPCR, EasyqpcR. We will choose HTqPCR because it is the most extended one. (Dvinge and Bertone, 2009)
HTqPCR – high-throughput qPCR analysis in R R package is designed for the analysis of cycle threshold (Ct) values from quantitative real-time PCR data. This insight is strongly related to the computation speciality, as it encompasses areas such as data analysis, data mining and statistics, amongst others.

Bioinformatics as a service Bioinformatics as a service is materialised in the creation of a web application for the aforementioned *HTqPCR* R package. This insight is related to software, and as this project is not software-oriented, the web has been deployed for merely practical reasons. The idea was to offer the functionalities included in the R package in a visual eye-friendly way, thus professionals from other areas such as biology without any prior programming knowledge could use it. For these reasons, web development methodologies, implementation and management related documentation will be out of the scope of this project.
Consequently, the focus within this dissertation will be on bioinformatics as a research field complementing it with the developed web.

1.3 Project management

1.3.1 Project WBS – Work Breakdown Structure

TABLE 1.1: Project WBS

Project					
Web		**Dissertation**		**Research**	**Management**
Design	**Implementation**	**Design**	**Development**		

As this project work breakdown structure states, this bachelor final project work consists on doing a whole project, whose time is distributed in developing a web application, a dissertation, doing some research on qPCR data analysis and management tasks. The web application and dissertation must be first designed and developed.

1.3.2 Project time estimation

TABLE 1.2: Summary of project time estimation. (See A for a more detailed project time estimation)

Tasks	**# Hours**	**Percentage**
Fundamental research	20	6.67%
Management	10	3.33%
Web design	10	3.33%
HTqPCR research and web development	200	66.67%
Dissertation design	5	1.67%
Dissertation development	55	18.33%
Total number of hours	300	

This project of 12 ECTS credits consists of 300 total hours of work. Research should take an average of 50% of the whole project, another 20% is needed to write this dissertation and the remaining 30% comprises the rest of the tasks: web design, implementation and project management.

Chapter 2

Fundamental biological concepts

2.1 DNA strands and proteins

As part of this project scope, in this section, DNA strands (Nelson and Cox, 2012) will be introduced in order to understand where the qPCR data that it is being analysed in this dissertation comes from.

DNA makes us unique, this molecule does not only difference us from other life beings but it is also related to how we are. How is that incredibly possible?

DNA strands are a type of nucleic acids, which are made of smaller molecules called nucleotides. Nucleotides are composed of three molecules: a pentose, a nitrogenous base and a phosphate group, as can be seen below:

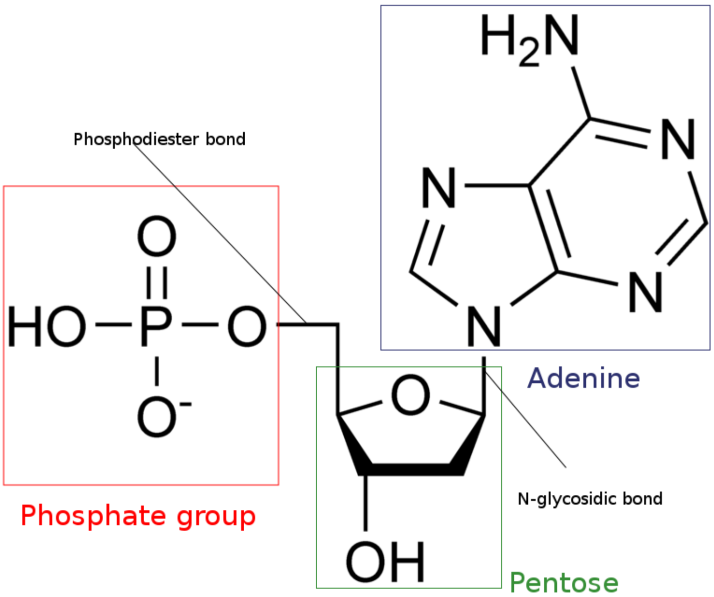

FIGURE 2.1: Adenine-based DNA nucleotide

This is deoxyadenosine monophosphate (DNA nucleotide), a nucleotide based on a phosphate group, a pentose called deoxyribose in the case of DNA nucleotides, and a nitrogenous base named adenine. The phosphate group of the nucleotides and the pentose remain the same in the case of the DNA nucleotides. Therefore, the nitrogenous base gives the identity to them.

There are four types of nitrogenous bases regarding DNA nucleotides: adenine (**A**), guanine (**G**), cytosine (**C**) and thymine (**T**).

Taking into consideration their nitrogenous bases, there are four types of DNA nucleotides: deoxyadenosine monophosphate (**dAMP**), deoxyguanosine monophosphate (**dGMP**), deoxycytidine monophosphate (**dCMP**) and deoxythymidine monophosphate (**dTMP**).

To sum up, in IT terms, the DNA strand can be seen as a character array made up using 4 characters: 'A' of adenine, 'G' of guanine, 'C' of cytosine and 'T' for thymine.

These DNA nucleotides are linked using phosphodiester bonds between their phosphate groups to build a polynucleotide. According to the DNA double helix model, to build a DNA molecule as known nowadays, two antiparallel polynucleotides are needed taking into account the following bonds between nitrogenous bases called base pairs: **A**=**T**, **G**≡**C**. In order to stabilize the double helix structure, there are electrostatic and hydrophobic bonds between bases.

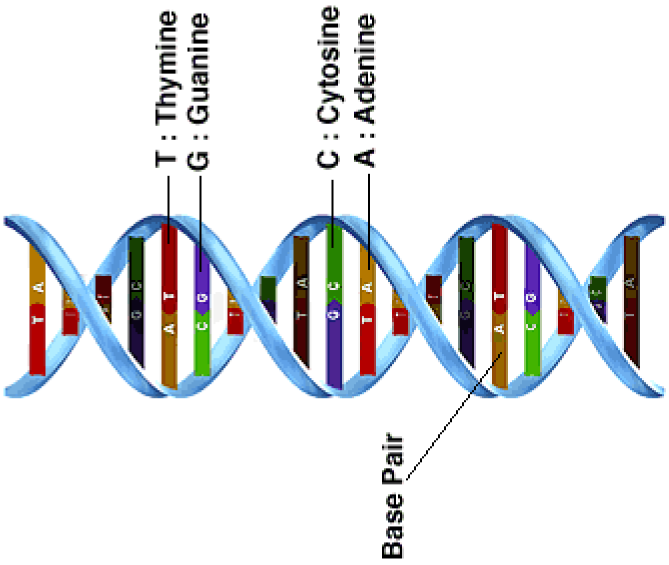

FIGURE 2.2: DNA double helix model

The complementarity between nitrogenous bases is crucial for the replication procedure that is carried out in the DNA strand. **Replication** is the generation of new DNA molecules using the existing DNA molecules as a mould, creating the complementary DNA strand using the respective base pairs. This is, essentially, what it is performed artificially in the qPCR experiment.

It has been explained previously that DNA strands can be represented as character arrays composed of just four types of nucleotides: **A**denine, **T**hymine, **G**uanine and **C**ythosine. However, what does the information contained in the DNA strand consist of? To understand this, we need to focus on another type of molecules: **proteins**.

The proteins are molecules composed of chains of smaller units called amino acids. As opposed to DNA strands, which have just 4 basic types of nucleotides, the proteins have more than 21 basic amino acids. Therefore, it is possible to build a wide variety molecules using proteins comparing to the molecules that the DNA can build.

Thanks to this variety, proteins are in charge of most of the life tasks. However, this would not be possible without the DNA, because it brings the fundamental information to synthesise proteins. This information is typically organised on *genes*, segments that encode proteins.

2.2 PCR and qPCR

In the previous section, DNA nucleotides and the composition of a DNA molecule was introduced because PCR technology takes advantage of DNA replication to amplify DNA sequences. Since the qPCR data is researched, in this section, PCR and its derivative qPCR are going to be explained. (Bustin, 2012)

PCR

The polymerase chain reaction (PCR) is a technique used in molecular biology to amplify a small sample of DNA fragments using DNA's capacity for replication. It is useful to research genetic diseases, as well as surprisingly in the field of forensics to identify criminals. This method is based on replication cycles: in each cycle, the amount of DNA is duplicated.

PCR needs some basic components in order to work: the DNA sequence to amplify, two primers (complementary to the DNA sample, supporting the polymerase in order to get it attached to the DNA sample), Taq polymerase (high-temperature resistant, at around 70ºC, this synthesise new DNA strands), Deoxynucleoside triphosphates (dNTP, the individual elements that make the DNA) and some chemical reagents.

PCR is typically done in between 20-30 cycles. Each cycle consists in at most four steps. In the first cycle an initialisation step may be necessary and a regular cycle includes an a denaturalisation step, an annealing step and an extension/elongation step.

- Initialisation step: some polymerase need heat activation in order to work. This is done by heating the reaction until 94-96ºC (98ºC if extremely thermostable polymerases are used) for 1-9 minutes.

- Denaturalisation step: in this step the double helix structure of the DNA is broken in order to get it replicated. This is done by incrementing temperature until 94-98ºC for 20-30 seconds so as to break the hydrogen bonds between bases. At the end, single-stranded DNA molecules are achieved.

- Annealing step: primers get bonded to the single-stranded DNA sample by lowering the temperature until 50-65ºC for 20-40 seconds. They will help the polymerase to do its job.

- Elongation step: Taq polymerase works optimally in a temperature between 75-80ºC, which 72ºC are commonly used. In this temperature, Taq polymerase synthesise a new DNA strand complementary to the DNA sample by adding dNTP that are complementary to those in the DNA sample.

Moreover, the process of PCR can be divided in 3 stages: exponential amplification, leveling off stage and plateau.

- Exponential amplification: each cycle doubles the amount of DNA.

- Leveling off stage: reaction goes slow because the polymerase loses its activity, and the reagents become limiting.

- Plateau: there is no more DNA duplication because of a lack of resources.

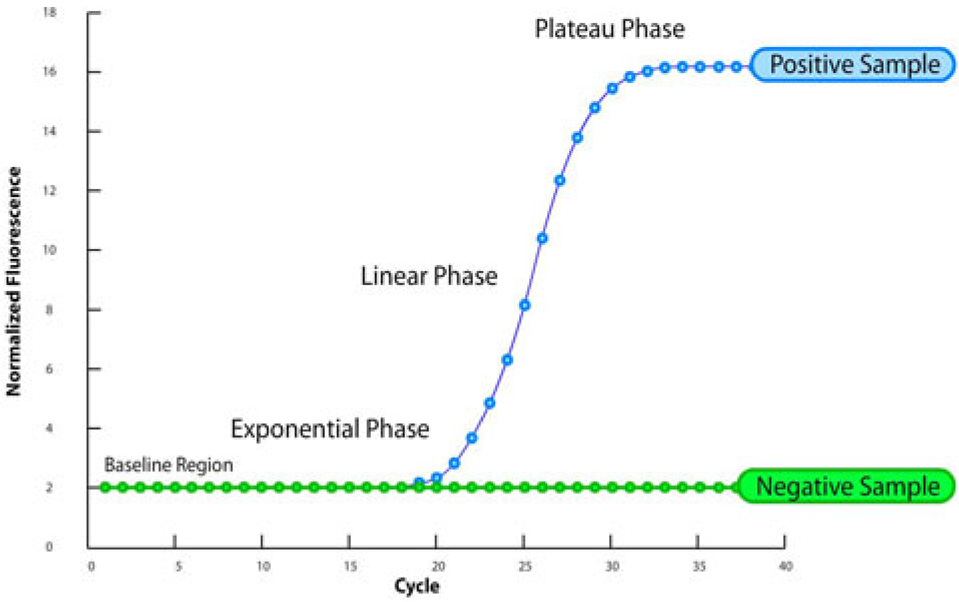

FIGURE 2.3: Typical PCR curve

qPCR

PCR has been previously explained, but the thing is how this reaction can be usted to measure the amount of DNA in a sample. This is used to determine whether a DNA sequence is present in a sample and the number of copies in it. It has a high degree of precision. Fluorescent dyes or fluorophore-containing DNA probes are used for that purpose.

For the qPCR, the generated DNA molecules have to be measured. This is achieved by taking advantage of the aforementioned fluorophores, which are chemical compounds that emit light when iluminated with UV light.

At the first cycles, the amount of DNA is too low to see light. However, once the number of DNA molecules reaches a certain amount, the light starts to be visible.

qPCR expression data

qPCR data is the information that is gathered from a qPCR experiment, which aims to measure the amount of DNA in a sample. However, what are we trying to quantify? How much DNA do we have in a specific cycle? In which cycle did we reach the plateau? In which cycle has the growth been detected by the fluorescent dye (light starts to be visible) and started to grow exponentially?

qPCR can be used to measure the aforementioned information. In this dissertation, we are going to focus on one of them: qPCR **expression** data. We want to know how much a gene is "expressed" in each of the samples. In other words, the amount of DNA molecules in the original sample is measured based on the number of cycles (i.e. duplications) needed until the light is visible. This is called cycle threshold, also abbreviated "Ct".

Cycle threshold is, technically speaking, the intersection between an amplification curve and a threshold line (See 2.4). ΔRn refers to the intensity of the fluorescent dye in real-time. Cycle number is the current cycle in the qPCR experiment. A threshold line is marked for the qPCR experiment, which triggers when the DNA starts to become detectable by the fluorescent dye. This threshold line should be the same for every amplification curve in a qPCR experiment so as to make sure that the Ct is calculated in the same way for every DNA sample inside the experiment.

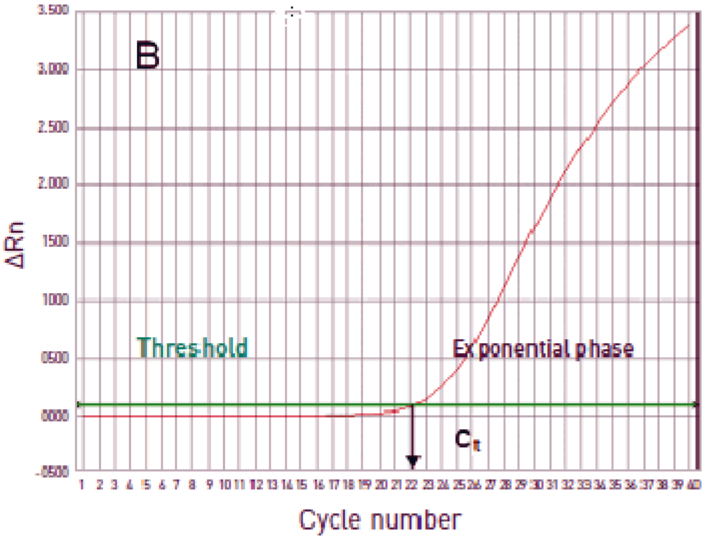

FIGURE 2.4: Ct amplification curve with threshold

How is this related to the DNA expression?

When Ct triggers because it has reached threshold means that it becomes detectable. Imagine that this Ct value is low, it means that the DNA has become detectable early, thus the initial amount of DNA is huge and it could be noted that this DNA sequence is highly expressed compared to another Ct value which is higher. This would mean that it has started to be detectable by the fluorescent dye later.

Different technologies, different formats

A general theory of a qPCR experiment has been previously explained but in practice there are different vendors that use different technologies, such as Applied Biosystems, Inc., Roche Applied Science, Bio-Rad or BioMark.

The introduced case covers only the process of what happens with one DNA sequence, but these vendors have develop systems to measure the amount of several DNA sequences simultaneously. For example, in the case of TaqMan Low Density Arrays (TLDA), propietary format of Applied Biosystems, Inc., DNA samples are located inside wells within a card. These cards could have 96 or 384 wells, capable of locating 96 or 384 DNA sequences.

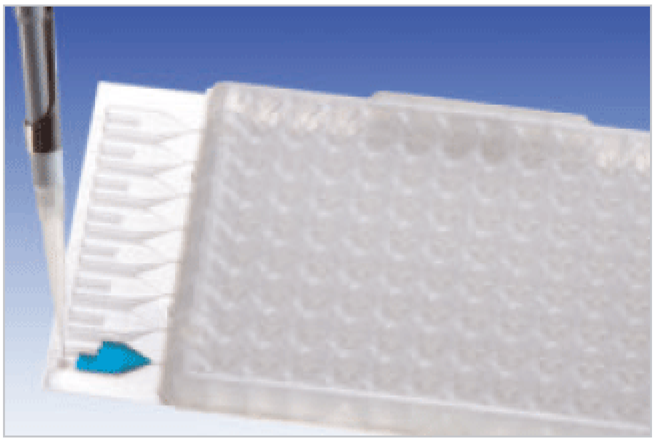

FIGURE 2.5: TLDA card

However, there are also non-well based microfluidic systems, which are capable of working in two dimensions: DNA samples and for example, different organisms' blood. So you can get how much of every DNA sample is expressed for each of the organisms. This is used when massive qPCR data is aimed to be gathered.

Every format and technology have their pros and cons and they depend entirely on which kind of experiment is needed.

Chapter 3

Fundamental statistics concepts

In this chapter basics of statistics related to our bioinformatics application will be reviewed, such as some concepts of descriptive statistics, estimation and clustering, which are going to be used afterwards to explain some of the functionalities of the qPCR data analysis. (Ewens and Grant, 2005)

3.1 Measures of central tendency: mean and median

Arithmetical mean

In descriptive statistics, arithmetical mean (or just, mean) is the sum of all measurements divided by the number of observations.

Mathematically, given a dataset which contains the following $d_1, d_2, \ldots, d_n$ values. The arithmetical mean of these values, $\overline{d}$, would be:

$$\overline{d} = \frac{1}{n} \sum_{i=1}^{n} d_i \tag{3.1}$$

This value is skewed by extremely large or small values in the dataset, and, thus, if we are trying to figure out a good representative of the values in the dataset median could be more appropriate.

Median

In descriptive statistics, the median value is the value that divides the higher half of a data sample from the lower half. Mathematically, given a ranked dataset which contains the following $d_1, d_2, \ldots, d_n$ values. The median of the dataset, $\tilde{d}$ would be:

$$\tilde{d} = \begin{cases} d_{\frac{(n+1)}{2}}, & \text{if } n \bmod 2 \neq 0 \\ \frac{1}{2} \times (d_{\frac{n}{2}} + d_{\frac{n}{2}+1}), & \text{otherwise} \end{cases} \tag{3.2}$$

This value, as opposed to the arithmetical mean, is not skewed by extreme values, so it is a better representative value for the dataset when it has outliers.

3.2 Measures of dispersion

3.2.1 Sample variance and standard deviation

Sample variance and standard deviation are used in statistics to measure how far a set of values are scattered from their mean in a sample. Sample variance is the square of the standard deviation, which are represented respectively by s^2 or σ^2 and s or σ.

They are represented mathematically as below:

Given a dataset which contains the following $d_1, d_2, \ldots, d_n$ values.

- Sample variance:

$$\sigma^2 = \frac{1}{n-1}\sum_{i=1}^{n}(d_i - \overline{d})^2 \tag{3.3}$$

- Standard deviation:

$$\sigma = \sqrt{\sigma^2} \tag{3.4}$$

3.2.2 Coefficients of variation

Coefficients of variation are relative variation measures between the sample standard deviation and its mean. They are commonly represented as percentages. The coefficients of variation are useful because they are dimensionless, they help to compare deviation between two variables, which may have different means and measure units. Given the $d = d_1, d_2, \ldots, d_n$ dataset, it is defined mathematically as below:

$$CV = \frac{\sigma}{|\overline{d}|} \tag{3.5}$$

3.2.3 Interquartile range (IQR): quartiles and percentiles

In descriptive statistics, a percentile is a measure to indicate the value below which a given percentage of observations in a group of observations can be found.

For example: the 35th percentile (P_{35}) is the value below which 35% of values are found.

Quartiles are a subset of the percentiles, which comprise the values that divide a ranked data set into four equal parts. The first quartile(Q_1 or P_{25}), or splits the lowest 25% of the data from the 75% one. The second quartile(Q_2 or P_{50}) is the median and it splits the dataset in half. The third quartile(Q_3 or P_{75}) is the opposite to the first quartile: it splits the 75% of the data from the remaining 25%.

There are different methods in order to calculate percentiles. The most common one will be introduced: *The Nearest Rank* method.

Mathematically, given a ranked dataset which contains the following $d_1, d_2, \ldots, d_n$ values. Using *The Nearest Rank* method, P_i is calculated as below:

$$P_i = d_k \text{ where } k = \frac{i}{100} \times n \tag{3.6}$$

The interquartile range (IQR) is the difference between Q_3 and Q_1 giving us an idea of spreadness. It is used to draw boxplots.

3.3 Estimation

In this section, significance tests about hypothesis will be introduced. As this section will make difference between population and samples, statistics will be used to describe samples and parameters to describe population.

3.3.1 Significance tests about hypothesis: hypotheses, test statistics and P-value

The objective of statistical hypothesis tests is to test a hypothesis, based on a sample or a dataset. Testing a hypothesis consists on deciding whether it is asserted or not.

In statistics, a hypothesis is a statement that tells that the parameter of a population takes a singular value (H_0, null hypothesis). On the contrary, the alternative hypothesis(H_a) is the hypothesis against the null hypothesis.

Therefore, there are four possibilities, regarding the assertion of the null hypothesis:

1. H_0 is true and it is asserted: right decision.
2. H_0 is true and H_0 is rejected: Type I error.
3. H_0 is false and H_0 is rejected: right decision.
4. H_0 is false and H_0 is asserted: Type II error.

α is the probability of type I error and it is called significance level since it measures how much strange or extreme data is to reject H_0. The most common significance levels are between 0.05 and 0.01.

There are lots of statistical test so as to validate different hypotheses. Therefore, depending on the aim, one of them will be chosen. Moreover, there are two sorts of test, regarding their assumptions:

Parametric tests This kind of tests assume that the data follows a well-known probabilistical distribution.

Non-parametric tests This type of tests have generally more flexible assumptions and they do not state conditions for the data distribution.

A statistical test uses its own test statistic, which is a standardised value that is calculated from sample data during a hypothesis test. It measures the degree of agreement between a data sample and the null hypothesis.

Once the test statistic is determined, this is used to calculate a p-value. This value is the probability of randomly achieving sample data or more extreme/strange than it. Therefore, when the p-value is low, being H_0 true, it means that the probability of getting this kind of data is low and thus, H_0 may be rejected if it is below the significance level. On the other hand, if the p-value is high, there is a strong evidence around the null hypothesis. In that case, it is concluded that H_0 can't be rejected.

However, what happens when more than one test are done?

α probability is just bonded to one test, so when more than one test are done, the type I error gets bigger. In order to control α, p-values or significance levels are adjusted using different methods such as Bonferroni correction or Benjamini & Hochberg.

3.4 Clustering

By definition, clustering is the unsupervised classification of patterns, (Tan, Steinbach, and Kumar, 2005) which of them could be observation or data elements, into groups which are called clusters. Its main purpose is to structure unlabelled data and as it unsupervised, there is no training/test data as it is common on supervised classification methods in the field of machine learning.

There are different types of clustering: partitional clustering, hierarchical clustering, exclusive clustering. As part of this project, hierarchical clustering will be analysed in the next section.

3.4.1 Hierarchical clustering

Hierarchical clustering is a method of cluster analysis that aims to arrange elements as being "above", "below" or "at the same level" one another building clusters. Being $k \in 1 \leq k \leq n$ and n the number of elements in the data, there are generally two types of hierarchical clustering procedures:

- "Bottom up" approach: at the beginning, every element is a cluster. After, two by two, clusters are merged into one until k clusters remain. In every iteration, the nearest two clusters are merged.
- "Top down" approach: at the beginning, all elements are inside a single cluster. After, cluster is divided in two until k clusters remain.

Moreover, a metric (distance between elements) is needed and a linkage criterion which specifies how distance between two clusters is measured. On the one hand, distance between pairs of elements can be measured using e.g. Euclidean distance, Manhattan distance or any metric that aims to calculate how far an element is from the other. On the other one, common linkage criteria comprise single, complete, average and centroid linkage.

- Single linkage: the nearest two cases of the clusters are taken to calculate distance between them.
- Complete linkage: the farthest two cases of the clusters.
- Average linkage: the mean between the cases of the clusters.
- Centroid linkage: the distance between centroids of the elements of the clusters.

Those chosen metric and linkage criterion would have an impact on the clustering result.

Chapter 4

Analysing qPCR data and web development

4.1 Preliminaries and workflow of qPCR data analysis

The fundamental concepts have been previously analysed in order to understand what it is going to be next: qPCR expression data analysis. HTqPCR is the chosen R package that will be used to analyse data. However, before we start to analyse it, the package functionalities that were integrated in the web and a general workflow will be introduced, which we will follow in this dissertation to analyse the sample data that is provided.

Functionalities

The R package HTqPCR provides functions to import data, assess quality, normalise, visualise data and test for statistical significance in Ct values between different features(i.e genes, DNA sequences). In this section, a summary of the features the package provide us is included and afterwards in the next sections, each feature will be thoroughlier analysed and explained with an example.

As written in the project definition, a web has been developed for practical reasons integrating some of the features included in HTqPCR. They are divided in two groups, data visualisation functions and data analysis functions:

TABLE 4.1: Features of the HTqPCR package

Data visualisation functions	**Data analysis functions**
Raw visualisation	**Filtering**
Overview of Ct across groups	Setting categories
Spatial layout	**Normalisation**
Duplicated features within samples	Quantile normalisation
Variation within/across samples	Rank invariant scaling
Quality assessment	Rank invariant normalisation
Correlation between samples	deltaCt normalisation
Distribution of Ct values	Geometric mean normalisation
Comparison of Ct values for two samples	**Data clustering**
Scatter across samples	Hierarchical clustering
Ct heatmaps	Principal components analysis
Coefficients of variation	Differential expression
Fold changes	t-test
Relative quantification	Mann-Whitney
Detailed visualisation	

Workflow of qPCR data analysis
In order to achieve the goal of analysing qPCR data, the following schema will be followed:

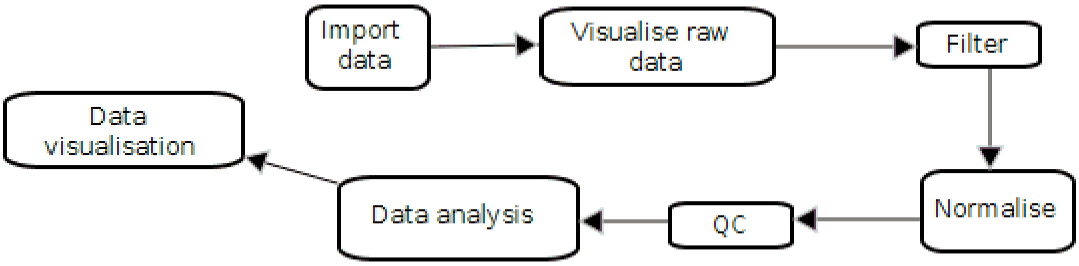

FIGURE 4.1: Workflow of qPCR data analysis

Loading data The only requirement to load data for the HTqPCR package is that it should contain columns of Ct values and feature names information. The web application has more requirements. For more information about the qPCR data and the web application requirements, see 4.9.3.

Quality of the raw data Once data is loaded, the quality of the input data must be checked using raw visualisation functions. There may be multiple problems with the data generation and removal of any bad sample could be needed.

Preprocessing for analysis Filter and normalise data so as to analyse it. Filtering is needed for setting categories in case of missing, out of range values or meaningful deviation. There are a bunch of normalisation methods available in order to make data comparable.

Quality assessment Once the data is filtered and normalised, check its quality again.

Data analysis Look for differences, cluster samples and/or genes or make predictions.

Data visualisation Fold changes are provided to analyse differential expression results.

This workflow is what is done in general, however, it is possible to e.g. use quality assessment before or/and after normalisation in order to look at interesting changes or filtering before or/and after normalising. It is important to think carefully of what it is wanted to analyse and achieve.

4.2 Visualising qPCR raw data

4.2.1 Overview of Ct across groups

It might be interesting to get a general overview across samples of the values of some features before starting to process data. A general overview of all features is possible to get, but it wouldn't be so meaningful because the chart would look overfull.

The chart introduced calculates the mean for each specified feature and the 97.5% confidence interval related to each of them using a t student distribution. Features might have replicates across and within samples, they all will be taken into account when doing the computation. Confidence interval calculation gives the chance to know how much Ct values deviate from each other for a certain feature. The sum of feature replicates across samples is regularly small and as it is tried to describe samples drawn from a full population, t student distribution is used.

In case a calibrator is chosen, a log2 ratio is plotted, showing the Ct values relative to the chosen calibrator. This is done by calculating the mean of the calibrator features and dividing the mean of the chosen features to the mean of the calibrator features.

The parameters of this chart are the following ones:

- Features: Ct values of a set of interested features that are wanted to plot.
- Samples: samples to take into the account for the calculation.
- Groups: samples are divided by groups, each of them having its proper calculation.
- Calibrator: calibrator feature type, every feature of this type is selected as calibrator features.
- Confidence interval: should confidence interval be plotted?
- Legend: should a legend be plotted? In which position?

Given the following example:

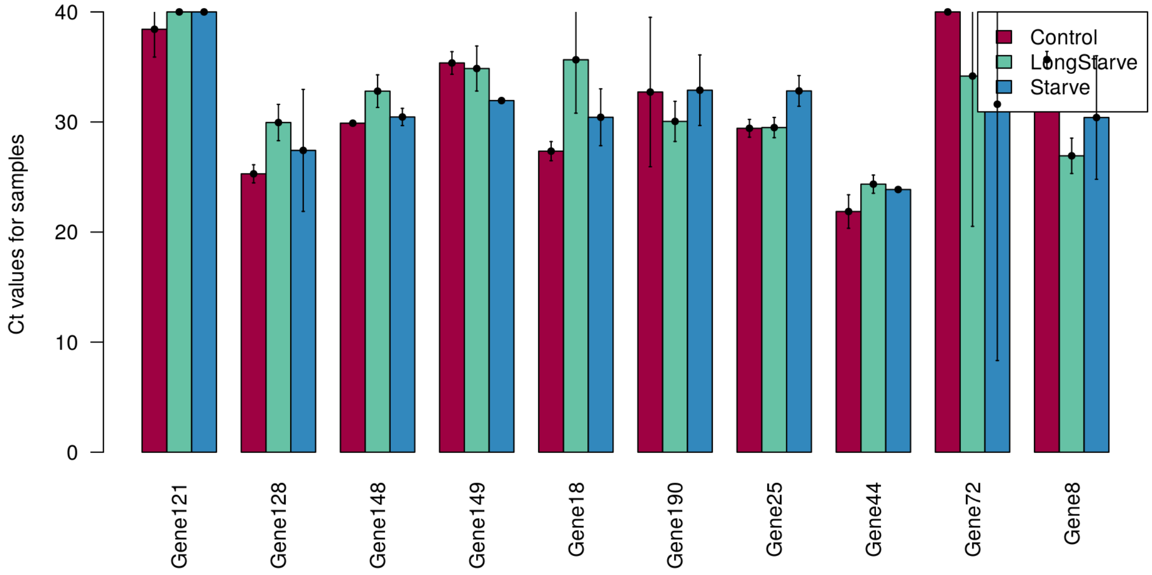

FIGURE 4.2: Overview of Ct across groups

Let's analyse "Gene121", it has two replicates within a sample and the following Ct values:

```
        sample1 sample2 sample3  sample4 sample5 sample6
Gene121      40      40      40 36.79387      40      40
Gene121      40      40      40 36.90377      40      40
```

For each of groups (Control, LongStarve and Starve), mean is calculated. For LongStarve and Starve, mean remains 40 because all values are equal but for Control, mean between the values of sample1 and sample4 are calculated, which is 38.42441.

```
 sample1  sample2  sample3  sample4  sample5  sample6
40.00000 40.00000 40.00000 36.84882 40.00000 40.00000

$Gene121
   Control LongStarve     Starve
  38.42441   40.00000   40.00000
```

Let's calculate confidence intervals for each of the groups:
First standard deviation for each of the groups is calculated,

```
$Gene121
   Control LongStarve     Starve
  1.819886   0.000000   0.000000
```

After that, t-score for the standard t distribution is looked for, in this case our sample length is 4 for all the groups and our confidence level is 0.975 so the same values for each of the groups is achieved:

```
$Gene121
   Control LongStarve     Starve
  2.776445   2.776445   2.776445
```

Then, the margin of error is computed by calculating the standard error for a population mean:

```
$Gene121
   Control LongStarve     Starve
  2.526406  0.000000   0.000000
```

Finally, confidence intervals are achieved using $[\hat{p} - |err|, \hat{p} + |err|]$:
For Control: $[38.42441 - 2.526406, 38.42441 + 2.526406] = [35.898, 40.95082]$
For LongStarve and Starve: $[40 - 0, 40 + 0] = [40, 40]$

4.2.2 Spatial layout

The web application data structure requires that features are organised in a particular spatial pattern, such as 96- or 384-well based TLDA. This brings up the idea of having a chart to plot the Ct values or other characteristics of the features.

Spatial layout chart is inspired on the physical design of a TLDA card and it helps to have a general overview of all the features in one specific sample as well as having it in its original physical structure. The main disadvantage of this chart is that only one sample can be plotted at a time. By default, it is inspired on a (16x24) 384-well based TLDA.

The parameters of this chart are the following ones:

- Sample: which of the samples to plot.
- Feature aspect: characteristic of the features to plot: ["Ct", "Feature type"]
- Number of rows of the layout.
- Number of columns of the layout.
- Ct range to colour: in case that Ct is chosen, the range between the colours blue and white are interpolated.
- Well size: the size of the wells.

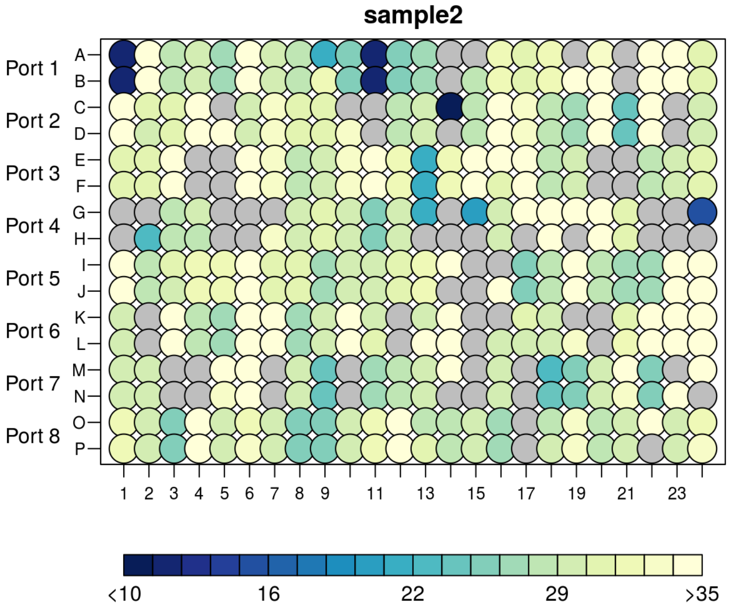

FIGURE 4.3: Spatial layout

The spatial layout is interesting to find out physical problems, i.e spatial patterns. For example, if a physical break over a row or a column is detected, the values in the spatial layout will show this pattern.

4.2.3 Duplicated features within samples

When the replications of each feature within samples is equal to 2, it is possible to have an overview of the features deviation just plotting one replicate against the other.

This chart gives the opportunity to have a general overview of the features' deviation of a certain sample. It simply plots one feature replicate Ct value against the other one. When the deviation is not meaningful, the feature will be near from the y=x diagonal line. The chart gives the possibility to set a certain percent and the features that differ more than it will be marked.

Tab "Details" is available to query the Ct values of the marked replicates.

The parameters of this chart are the following ones:

- Sample: which of the samples to plot.
- Percent: the features that differ more than the typed percentage will be marked.

Given the next chart of duplicated features of sample 1:

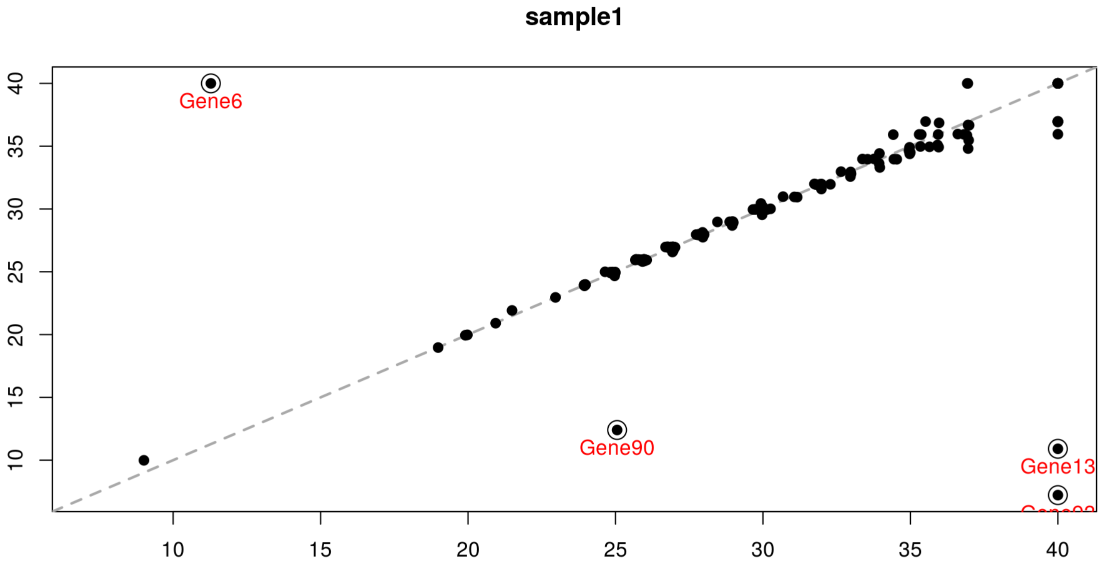

FIGURE 4.4: Duplicated features within samples

[“Gene6”, “Gene90”, Gene13”, “Gene93”] have been clearly marked because their values differ more than 20% in sample1.

Details tab shows the following:

```
Replicates differing > 20% on card 1:
           rep1      rep2
Gene13 40.00000 10.906260
Gene6  11.28030 40.000000
Gene90 25.05428 12.402942
Gene93 40.00000  7.217595
```

In quantitative science, the term “percentage difference” shows a difference between two values as a percentage when both values mean the same (one value is not older or better than the other).

Given two numerical quantities, x and y and their difference $\Delta = |x - y|$:

$$\%Difference = \frac{\Delta}{mean(x, y)} \times 100 \tag{4.1}$$

For example, the percentage difference between Ct values of “Gene90” is:

```
  Gene90
67.55085
```

Therefore, it is deduced that differs more than the 20% so it must be marked.

4.2.4 Variation within/across samples

Instead of having a general overview of a set of features across samples or all features in a specific sample, let's try to get a better overview of the data focusing on specific information of the data that may be more useful for us. In the previous section, Ct values were analysed when they are duplicated within a sample, but what happens when their replication amount is higher? And when comparison across samples is required? The amount of dimension increases, so it is not just as simple as plotting one value against the other.

Charts are provided to assess the variation of the feature replicates within and across samples. This is very useful to know if some samples or features are less reliable and need to be discarded. Reasons could be perhaps such as a high variation overall in the sample (Ct values of the feature replicates are very different between them) so the sample is discarded, or individual features within a sample show this high variation and need to be further investigated and discarded.

Two type of charts are available: a summarised boxplot or a detailed scatter plot. The first one gives a general overview of the variation of the samples, including outliers for each of them. The second one is aimed to further investigate outliers that had been identified.

Details for numerical variance/standard deviation and mean for each sample and feature replicates are provided as a table.

Parameters:

- Samples: samples to assess variation.
- Variation computation method: variance and standard deviation are available for calculus.
- Base 10 Logarithm: whether $\log$ scale should be used.
- Type of plot: summarised boxplot or detailed scatter plot are provided.
- Identify individual outliers: in case that detailed scatter plot is chosen, it is possible to add feature names to the plot.

Given the following summarised boxplot: (not from example data, it has been altered)

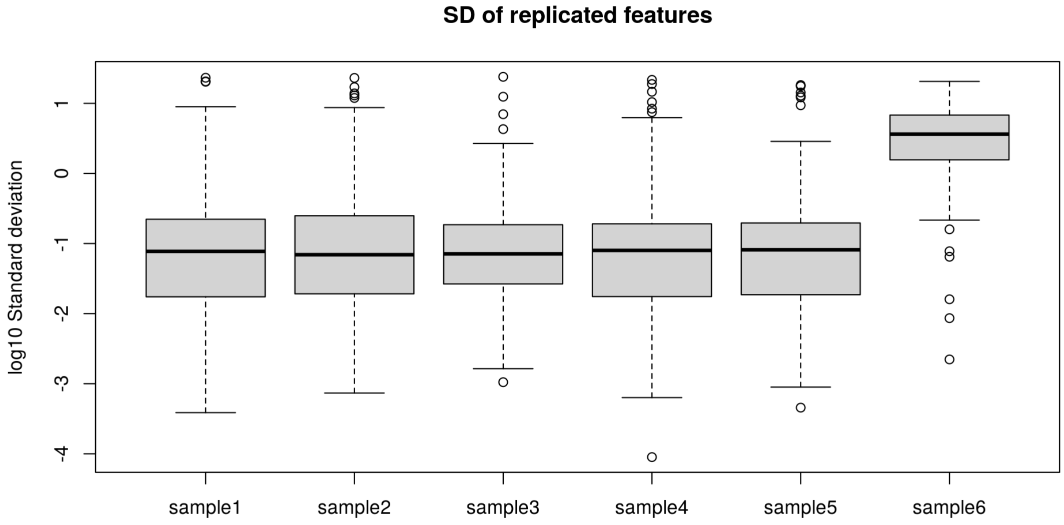

FIGURE 4.5: Variation within/across samples

It is obvious that feature replicates in sample6 highly vary and the sample should be discarded because what it is expected is that replicates within a sample have very similar values regardless of their position in the card. It is also noticeable that sample6 is very different from the other samples in the boxplot.

Given the following scatter plot: (not from example data, it has been altered)

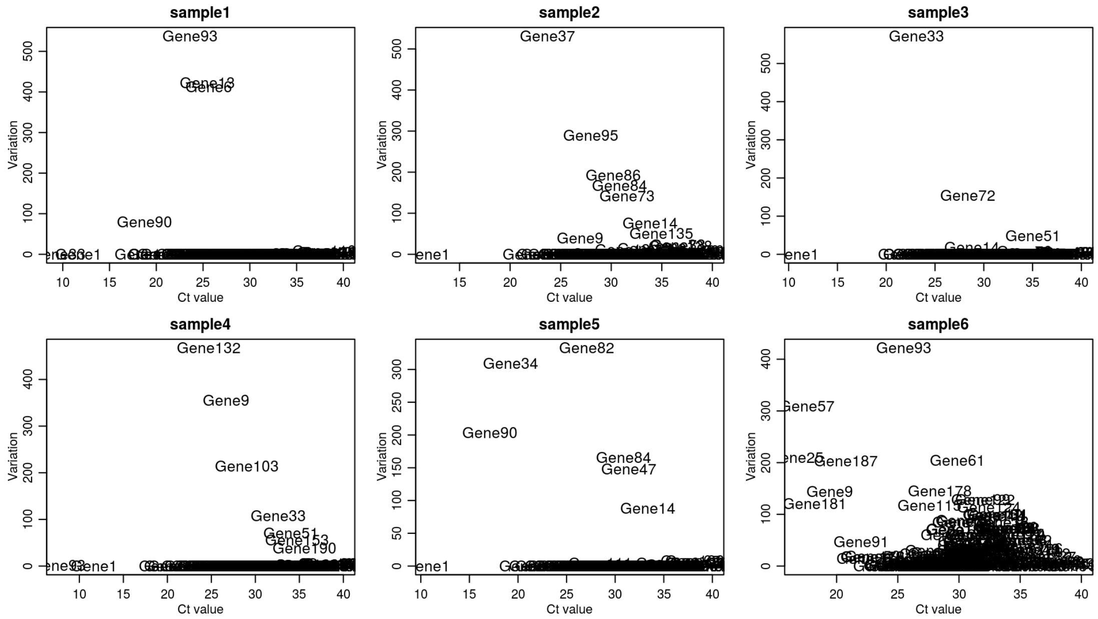

FIGURE 4.6: Variation within/across samples in detail

The highly variable feature replicates of sample6 can be clearly seen in this plot. It is expected that the variation between features tend to be zero, as it happens on the other samples. Individual outliers can be identified in the samples, in case they need to be further investigated and discarded from them.

4.3 Filtering data: setting categories

After visualising raw data, the next step in the workflow is filtering. Filtering is the process of removing redundant or unwanted information from the data. As it is stated in the qPCR data structure, features can have three categories: “OK”, “Undetermined” or “Unreliable”. It is possible to filter data i.e. setting categories depending on various criteria:

- Range of Ct values: High Ct values above a user defined threshold are not considered a reliable measure due to technical or biological reasons: they could mean that the DNA samples are not growing as fast as they are expected to grow, or they simply haven’t grown in the experiment. Low Ct values either not, for the inverse reason: they are growing too fast to be considered reliable. High Ct values are marked as “Undetermined” and low ones as “Unreliable”.
- Flags: When the experiment has been detected as “Failed” by the qPCR thermal cycler, it is marked as “Unreliable”.
- Variation within biological or technical replicates: Ct values of feature replicates outside a user-selected confidence interval (by default, 90%) within a sample or across samples of the same group are marked as “Unreliable”.

Taking this into account, in the web application, the following parameters are available:

- Max Ct: user defined upper bound. Every Ct value above this value will be considered a high Ct value and therefore, it will be marked as “Undetermined”.
- Min Ct: user defined lower bound. Every Ct value below this value will be considered a low Ct value and therefore, it will be marked as “Unreliable”.
- Accepted confidence interval: defined within $0 \leq x \leq 1$. The Ct values of features that are outside the accepted confidence interval will be marked as “Unreliable”.
- Grouping by: sets the grouping of samples so as to assess variation within feature replicates across samples of the same group.
- Flag to set unreliable: flag used by the thermal cycler to indicate that the experiment has failed for the computation of a Ct value.
- Collapse Ct from replicated genes w/n samples for standard deviation?: by default, variation feature replicates within a sample is assessed. This can be turned off.
- Consider flags?: If flags are taken into account to filter. If so, Ct values who have the flag to set unreliable, will be set so.
- Samples: samples to filter.

Once parameters are set, confirm those clicking on “Filter”: two charts are created and a details tab will show the results of the filtering in numbers in case that more details are needed.

In the next example, the next parameters have been set:

Max Ct: 35 , Min Ct: 10, Accepted confidence interval: 0.7, Flag to set unreliable: Failed, Collapse Ct from replicated genes w/n sample: yes, Consider flags: yes.

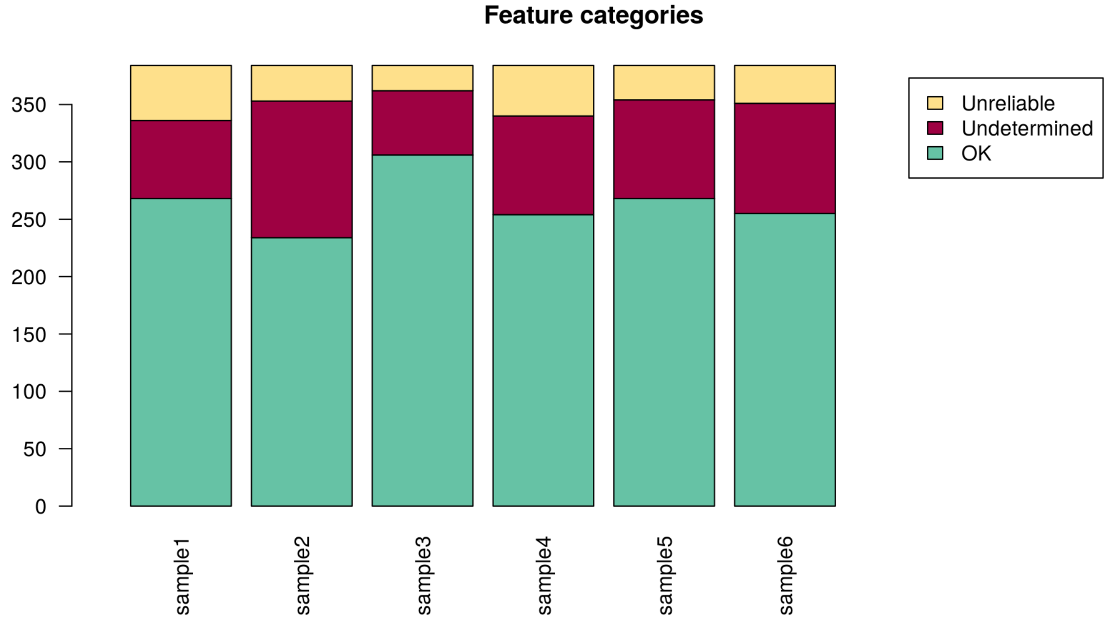

FIGURE 4.7: Filtering

```
            sample1 sample2 sample3 sample4 sample5 sample6
OK              313     264     327     295     296     286
Undetermined     68     119      56      86      86      96
Unreliable        3       1       1       3       2       2
Categories after standard deviation filtering:
             sample1 sample2 sample3 sample4 sample5 sample6
OK               268     234     306     254     268     255
Undetermined      68     119      56      86      86      96
Unreliable        48      31      22      44      30      33
```

4.4 Normalisation

Normalisation of the data is necessary before analysing it. It often happens that the conditions where data was actually measured are different. For example, in the case of the TLDA card samples, that different quantities of DNA are filled in the wells leading into a misunderstanding of the data.

The package offers different methods to normalise the data that was gathered. It is as simple as choosing the desired normalisation method and in case of deltaCt normalisation, choosing the control features. A plot of how each of the normalisation methods would remain is provided.

The normalisation methods that are available are the following ones:

- Quantile normalisation
- Rank invariant scaling
- Rank invariant normalisation
- deltaCt normalisation
- Geometric mean normalisation

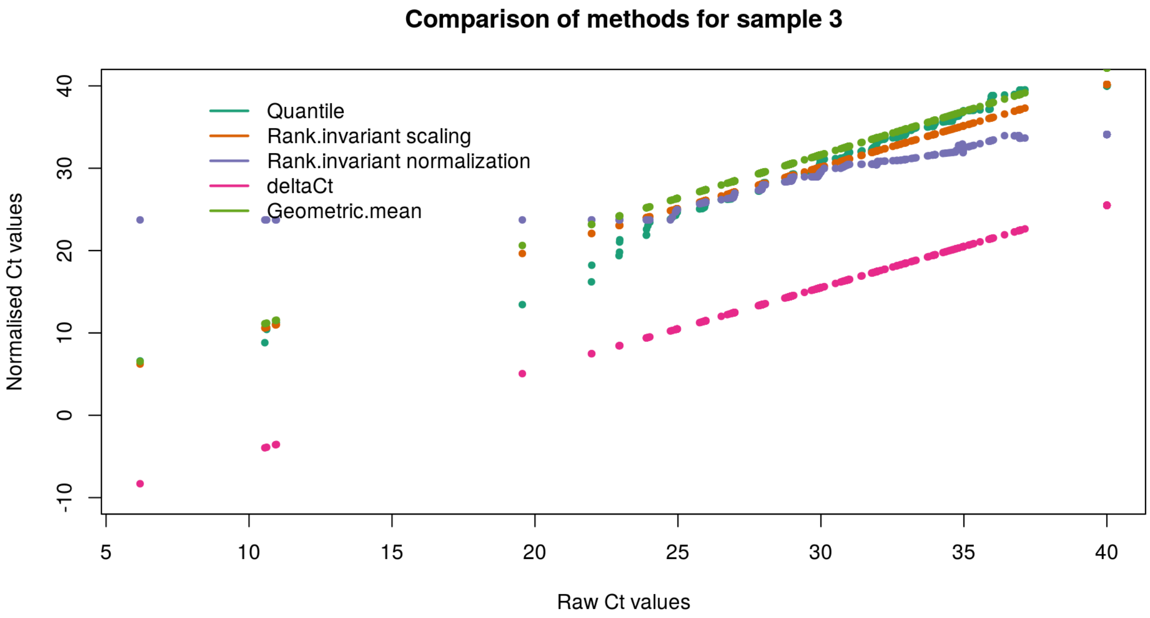

FIGURE 4.8: Comparison of normalisation methods

4.5 Quality assessment

4.5.1 Correlation between samples

Features within a sample could be correlated to feature of other samples, it means that changes in Ct values from one sample are accompanied by changes in Ct values of other samples. Taking this into consideration, it is expected samples within a group to be highly correlated and it is aimed to analyse correlation between samples of different groups to look for any interesting fact. In case that correlation between two specific samples is needed, it is better to use Comparison of Ct values for two samples chart.

The provided chart shows the correlation between samples using the Pearson correlation coefficients. Clusters are created hierarchically depending on how strong the correlation between samples is from the strongest to the weakest. By default, 1 minus the correlation is plotted.

Given the following example:

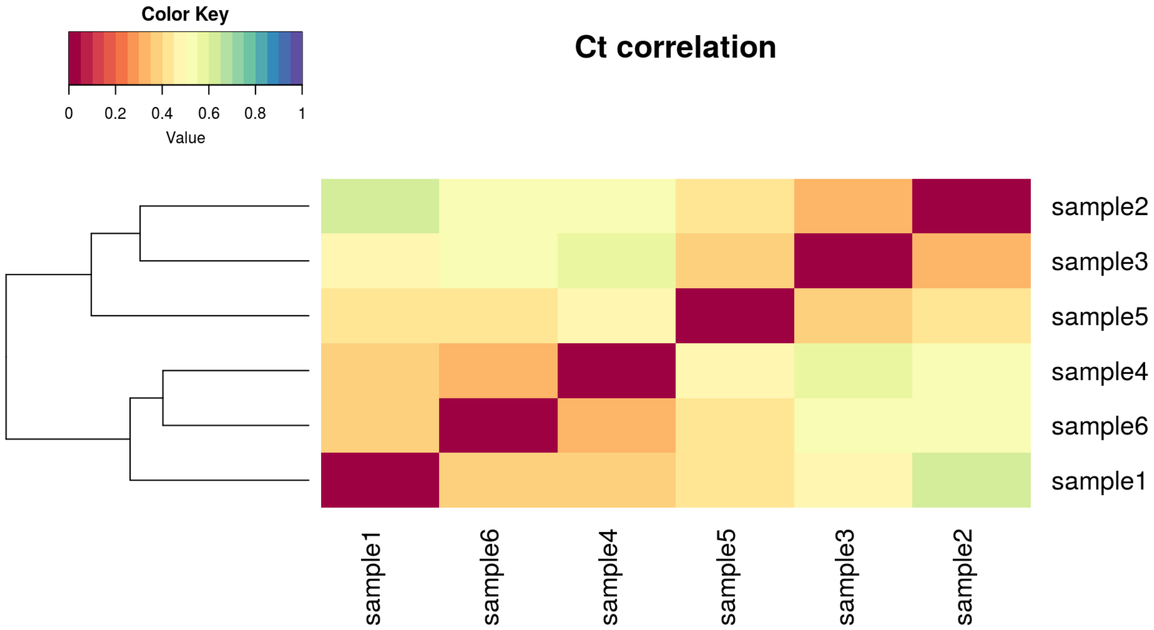

FIGURE 4.9: Correlation between samples

As expected, sample2 and sample3 have a high correlation because they are in the same group (LongStarve). Moreover, interestingly, there is a stronger correlation between sample4 and sample6, which are not in the same group, than e.g. between sample4 and sample1 or between sample6 and sample5.

4.5.2 Distribution of Ct values

Sometimes it is interesting to look at the sampling distribution of the data. Different tools are provided in the web application for this: a six-number summary (five-number summary including mean), density estimates, histogram and boxplots.

The six-number summary includes a summary for each sample of their min, Q_1, Q_2, $mean$, Q_3, max.

```
        sample1  sample2  sample3 sample4  sample5  sample6
Min.    " 7.218" " 7.408" " 6.19" " 6.853" " 6.787" " 5.133"
1st Qu. "26.738" "28.855" "27.90" "26.964" "27.913" "27.514"
Median  "28.937" "30.994" "29.92" "29.943" "30.778" "29.931"
Mean    "29.543" "32.190" "30.35" "30.590" "30.995" "30.663"
3rd Qu. "33.323" "35.985" "32.98" "34.694" "34.702" "35.046"
Max.    "40.000" "40.000" "40.00" "40.000" "40.000" "40.000"
```

The density estimate is based on Kernel density estimation (KDE). It estimates the probability density function of a continuous random variable based on sample data. It plots all the selected samples together.

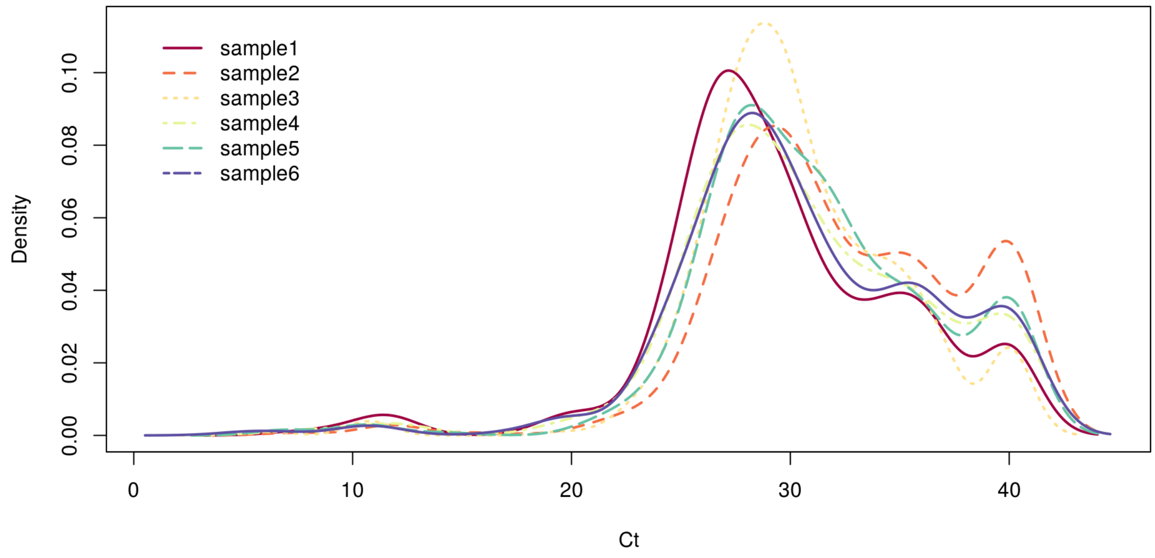

FIGURE 4.10: Density estimate of sample data

The histogram gives the possibility of plotting the frequencies of the individual Ct values of a single sample.

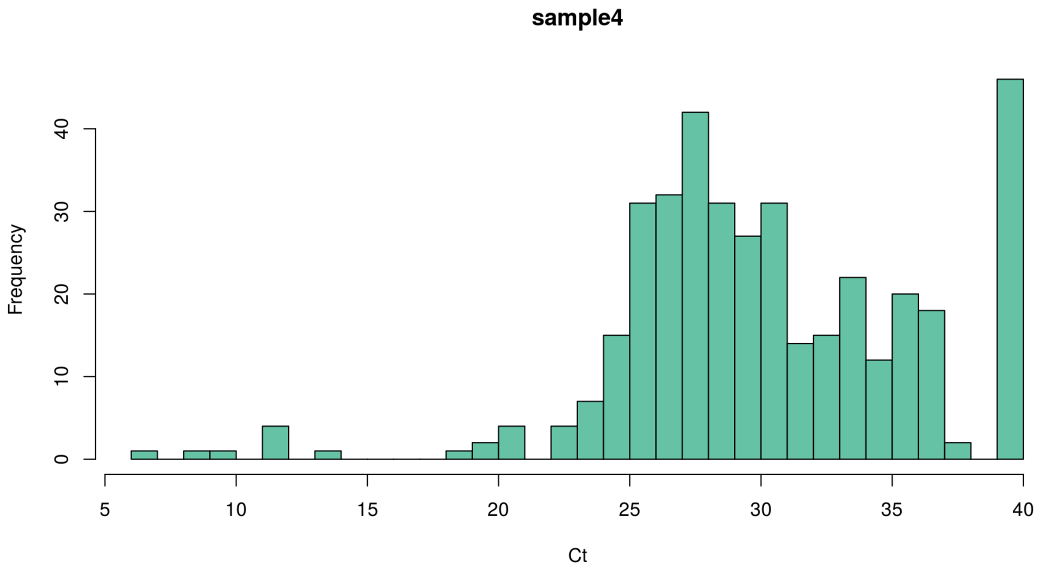

FIGURE 4.11: Histogram of sample 4

Finally, boxplots of Ct values across samples are provided. They can show possible outliers that could be in the data. They can be stratified by $featureType$.

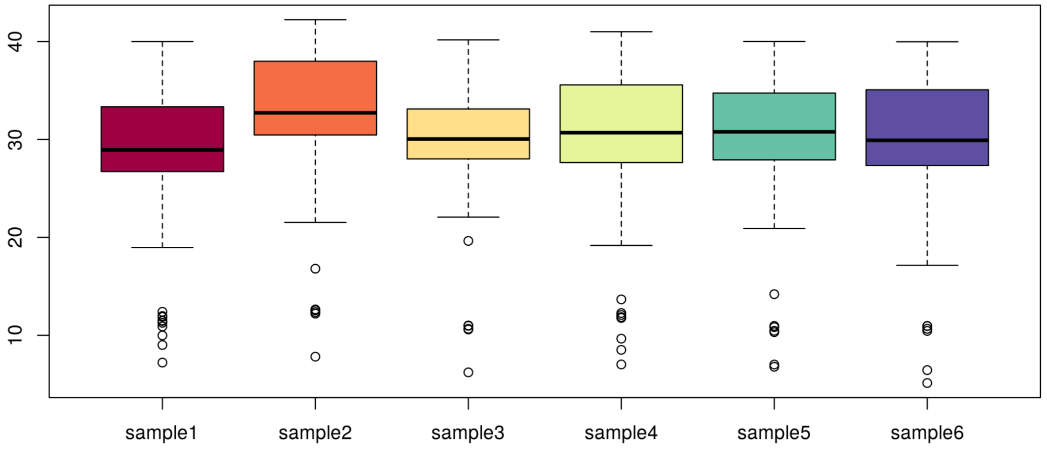

FIGURE 4.12: Boxplot of sample data

4.5.3 Comparison of Ct values for two samples

It is often needed to compare Ct values directly across two samples, this is possible to do considering that features in one sample and the other are the same. Therefore, the Ct values in one sample and the other will be plotted against so as to look generally for the variation of the Ct values of one sample and the other. A low variation between both would mean that they are very similar to each other. This is a good complement for Correlation between samples chart.

This chart provides, apart of plotting Ct values of samples against, the correlation information between both giving two Pearson correlation coefficient considering features below the specificied maximum Ct for correlation computation and considering no threshold at all.

The parameters of this chart are the following ones:

- The sample number 1.
- The sample number 2 to plot against.
- Y=X diagonal line: Should the diagonal line be plotted?
- Correlation info within plot: if correlation info between the samples would be plotted.
- Max Ct for correlation computation.

Given the next two examples below:

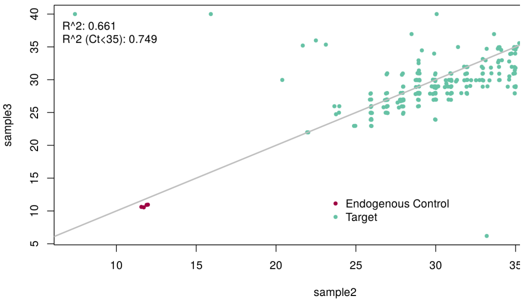

FIGURE 4.13: Comparison between two samples

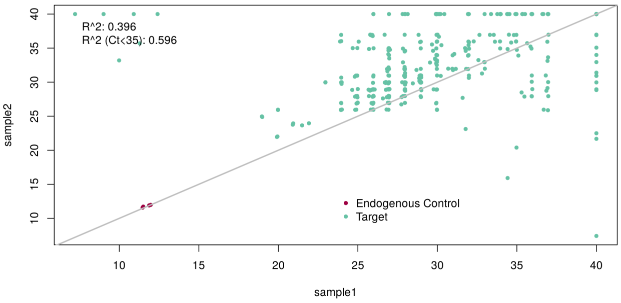

FIGURE 4.14: Comparison between two samples 2

When correlation between samples was analysed, it was mentioned that sample2 and sample3 shown a strong correlation, while sample1 and sample2 was weaker than expected, even if those were in the same group. Here it is possible to have a look with a bit more detail: points are nearer to the y=x diagonal line in the first case, so they have more similar Ct values, and farther in the second one.

4.5.4 Scatter across all samples

Using scatter across all samples, it is possible to compare Ct values directly across all samples, comparing them two by two. It is a generalisation for the Comparison of Ct values for two samples chart to do it across all samples.

Lower panels show the correlation information considering only features below the specified maximum Ct for correlation computation.

The parameters for this chart are the following ones:

- Samples: the samples wanted to compare across.
- Y=X diagonal line: Should the diagonal line be plotted?
- Max Ct for correlation computation.

Here an example:

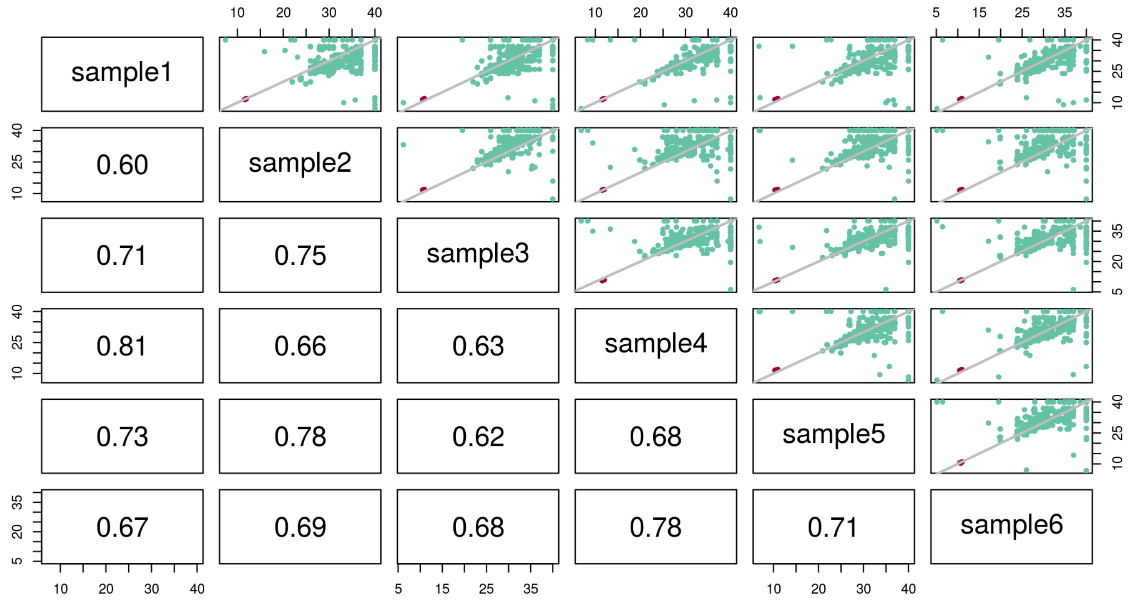

FIGURE 4.15: Scatter across all samples

4.6 Data clustering

4.6.1 Hierarchical clustering

It is sometimes useful to structure given data into groups, also called clusters, in order to look for a hierarchy of data elements specifying different criteria. Tree based models are visually attractive and meaningful when it comes to structuring things. Hierarchical clustering trees are provided inside the web application: features or samples are grouped by criteria such as euclidean distance, which focus on the magnitude of Ct values or Pearson correlation coefficient which focus on the similarities between them.

Thus, the parameters are the following ones:

- Clustering: what it is wanted to cluster: features or samples.
- Clustering by: specifies the metric used by the algorithm to calculate distances between the elements.
- The number of clusters aiming to mark with coloured borders. If nothing is specified, then no clusters are marked.

Here an example below:

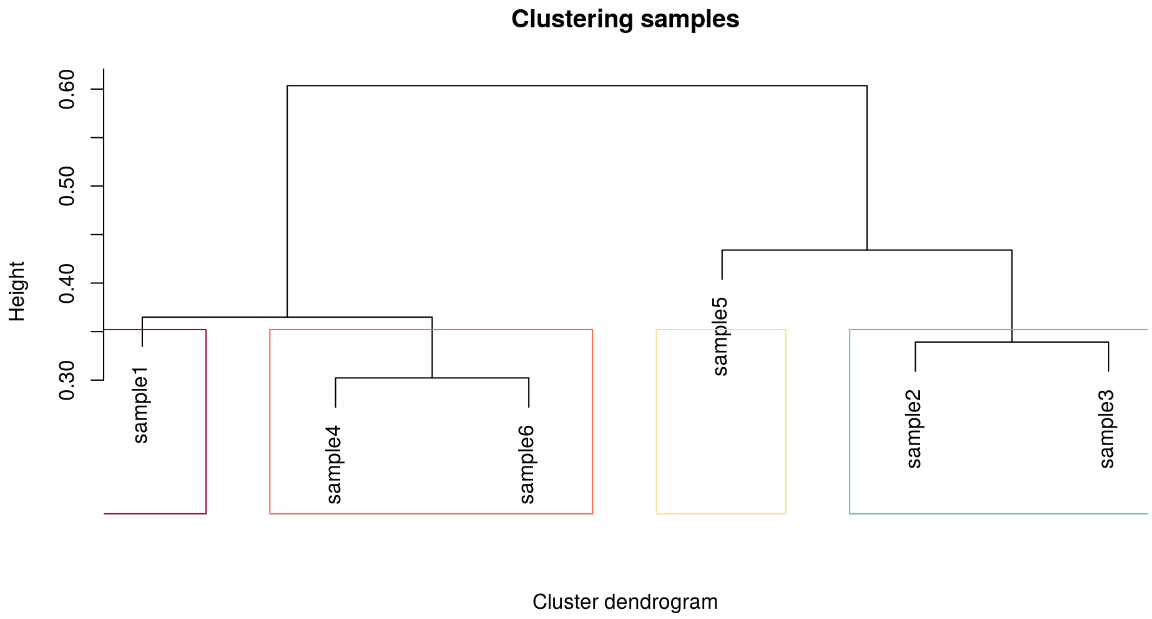

FIGURE 4.16: Hierarchical clustering

4.6.2 Principal components analysis

Principal components analysis extracts a new coordinates system for the data in which the variation of biggest size is captured in the first axis, called first principal component (PCA1), and the next one is captured in second axis, called second principal component. More variations are captured in the following dimensions, as components needed. It helps compressing lots of data into new dimensions that captures the essence of the original data.

In the web application, a principal components analysis chart is included. Features can be plotted along with the samples as a multivariate biplot, which aims to represent multivariate data on the same plot.

An example below:

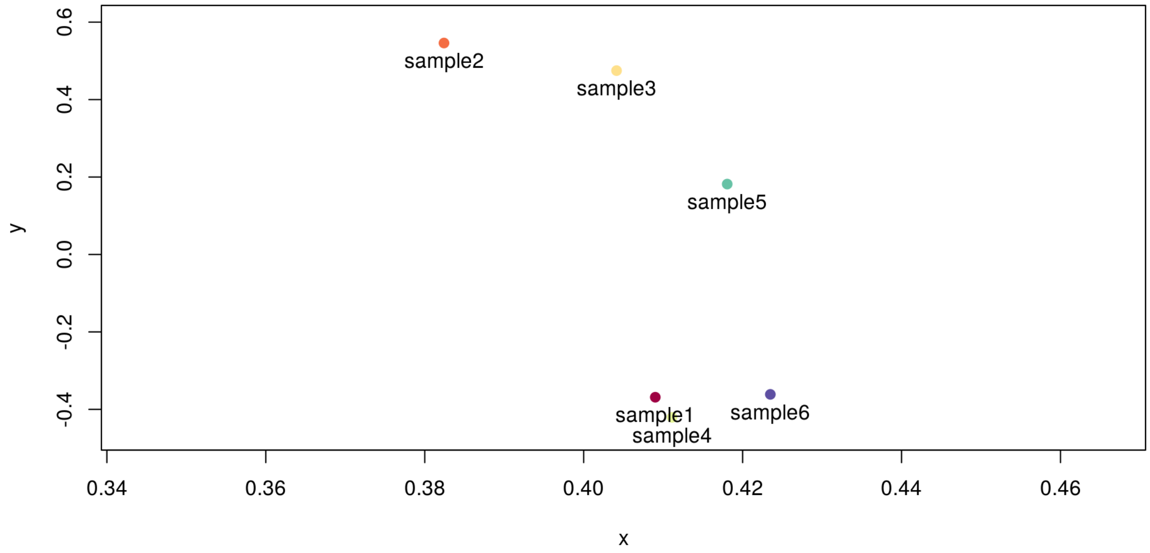

FIGURE 4.17: Principal components analysis

As the samples go farther (0,0) point, they gather the overall variation of the data. Moreover, in this example, it can be seen that all samples have a similar "contribution" to PC1, which is the component that comprises most variation in the data and two clusters can be clearly seen in the chart: one with sample1, sample4 and sample6 and another one with sample2, sample3 and sample5. sample1 and sample4 are in the same group (Control), as well as sample2 and sample3 (LongStarve). It could be noted that sample1 and sample5 are in the same group but not in the same cluster.

4.7 Differential expression

In this section, genes that are differentially expressed between different sample groups are aimed to be detected. There are two kind of tests in the package: t-test, a parametric test that assumes that the sample distribution follows the Student's t-distribution distribution and Mann-Whitney test, which is non parametric. Both tests compare two sample groups and state by default that the hypothesis is that the mean of the samples is equal in both groups.

4.7.1 t-test

t-test is a significance test based on hypotheses. As it has been explained in the section 3.3.1, a test statistic called, in the case of a t-test, t-value has to be determined in order to obtain P-values.

In the case of one-sample t-test, it is the following:

$$t = \frac{\overline{x} - \mu_0}{s/\sqrt{n}} \tag{4.2}$$

where:

$\overline{x}$ is the sample mean, s is the sample standard deviation and n the sample size.

However, t-values are calculated differently when there are more than one sample or when a paired t test is performed.

The parameters of t-test are the following ones:

- Grouping by: specifies how the samples should be grouped together.
- Compare 1: first group to compare.
- Compare 2: second group to compare first group against.
- A calibrator group between both of them, which is taken as a reference to the test.
- Hypothesis: by default it is two sided but it is possible to set it "greater" or "less".
- Use paired t-test: shall paired t test be used instead.
- Sort by p-values: states whether features should be sorted by their p-values.
- Flag unreliable/undetermined measurements as "Undetermined"?
- p-value adjustment method: "Benjamini & Hochberg" or "Bonferroni correction" are available to calculate adj.p.value.

Differential expression results should be saved in order to be visualised by fold changes.

4.7.2 Mann-Whitney

Mann-Whitney is a non-parametric significance test based on hypothesis. It is useful because it does not assume that the sampling distribution follows a normal distribution. As it has been explained in the section 3.3.1, a test statistic called, in the case of Mann-Whitney test, W-value has to be determined in order to obtain P-values.

In order to get the W-value the next procedure is performed:

1. Sample data is combined.
2. Rank all values: from the smallest to the biggest.
3. Calculate the mean of the ranks of each of the samples.
4. Calculate the absolute value of the difference between the values got in 3.

Once the W-values are determined, Mann-Whitney test uses the following formula to get a normal approximation:

$$Z_w = \frac{|W - \frac{m(m+n+1)}{2}| - 0.5}{\sqrt{\frac{mn(m+n+1)}{12}}} \tag{4.3}$$

Afterwards, p-values can be obtained.

Mann-Whitney test has the following parameters:

- Grouping by: specifies how the samples should be grouped together.
- Compare 1: first group to compare.
- Compare 2: second group to compare first group against.
- A calibrator group between both of them, which is taken as a reference to the test.
- Hypothesis: by default it is two sided but it is possible to set it "greater" or "less".
- Use paired test: shall paired test be used.
- Sort by p-values: states whether features should be sorted by their p-values.
- Flag unreliable/undetermined measurements as "Undetermined"?
- p-value adjustment method: "Benjamini & Hochberg" or "Bonferroni correction" are available to calculate adj.p.value.

4.8 Fold changes

Once differential expression results are saved, they can be later visualised using fold changes. The web application gives the possibility to visualise them using relative quantification and detailed visualisation.

4.8.1 Relative quantification

The relative Ct levels can be plotted between two groups. This helps to know more information about differences, whether they are significant or not.

In the web application, a chart for relative quantification is provided. It is possible to plot genes and their relative Ct levels between samples. Genes are marked with two different characters: "*" or "ïï" whether they are considered significant or very significant, respectively. The bars are hatched if the target and calibrator Ct are reliable.

The parameters of this chart are the following ones:

- Select differential expression result. If there are no saved results, fold changes can not be visualised.
- Choose features of interest.
- Transform: if Base 2 or 10 logarithm should be used.
- p-values' threshold: the maximum p-value for feature to be plotted.
- Cut-off for significant features: p-value threshold to consider a feature significant.
- Cut-off for very significant features: p-values threshold to consider a feature very significant.
- Mark significant feature: if features shall be marked by a significance level.
- Mark data with unreliable target o calibrator samples: if the bars are hatched with unreliable and calibrator samples.
- Include legend: to include a legend or not.

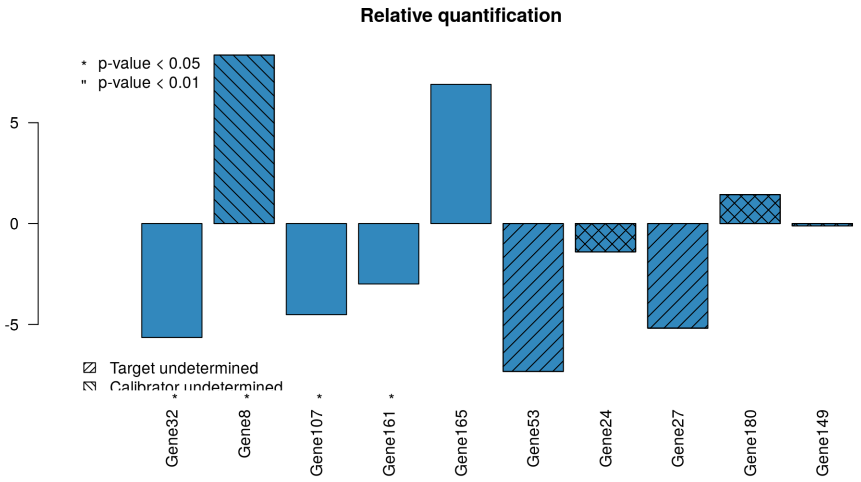

FIGURE 4.18: Relative quantification of some genes

4.8.2 Detailed visualisation

When more detail about fold changes is needed and relative quantification is not enough, the web application provides a detailed visualisation plot.

In this plot, the average of the mean calibrator samples and the average of the mean target samples are plot for each gene. To identify possible outliers, all individual Ct values of the genes are plotted (calculating the mean value of the replicates). Unreliable and undetermined Ct measures are coloured with red.

The parameters of the chart are:

- Select differential expression result. If there are no saved results, fold changes can not be visualised.
- Choose features of interest.
- p-values' threshold: the maximum p-value for feature to be plotted.
- Grouping by: specifies how the samples should be grouped together.
- A calibrator group between both of them, which is taken as a reference to the test.
- A test group between both of them, which is taken as the test.
- Cut-off for significant features: p-value threshold to consider a feature significant.
- Cut-off for very significant features: p-values threshold to consider a feature very significant.
- Mark significant feature: if features shall be marked by a significance level.
- Include legend: to include a legend or not.
- jitter: if Ct values are very similar, individual Ct values might lie on top of each other in the bars. This adds a jittering factor along the x-axis.

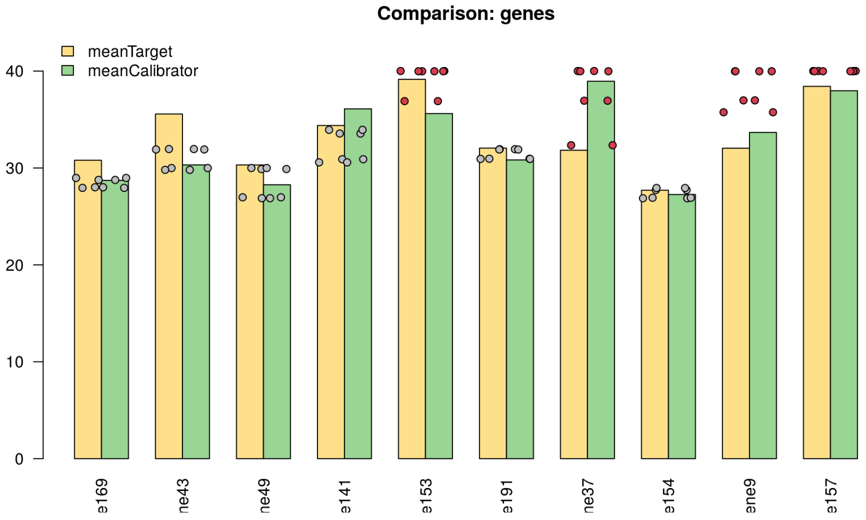

FIGURE 4.19: Detailed visualisation of some genes

4.9 Other web development aspects

4.9.1 Application structure

First of all, in order to get all the options in the web application, data needs to be imported. In the first page, two options are offered: it is possible to use the sample data that has been uploaded to the server or upload files. The files must fulfil the requirements stated in **qPCR data and files' structure**.

The application structure is pretty easy to follow, there is a navigation bar at the top in which a field of functionality can be selected such as accessing data, visualise raw data, etc. When one of them is selected, a second navigation bar may be shown that provides functionalities inside that field.

When it comes to making charts, the structure is quite standard, a sidebar layout is used. On the left side, there is the sidebar panel that includes parameters for the chart. On the centre and the right, the main panel that includes the chart. The main panel includes frequently a text area to include a title to the plot and saving functions in different formats. When more than one chart are produced or more details could be queried, a navigation bar is created.

Application file structure

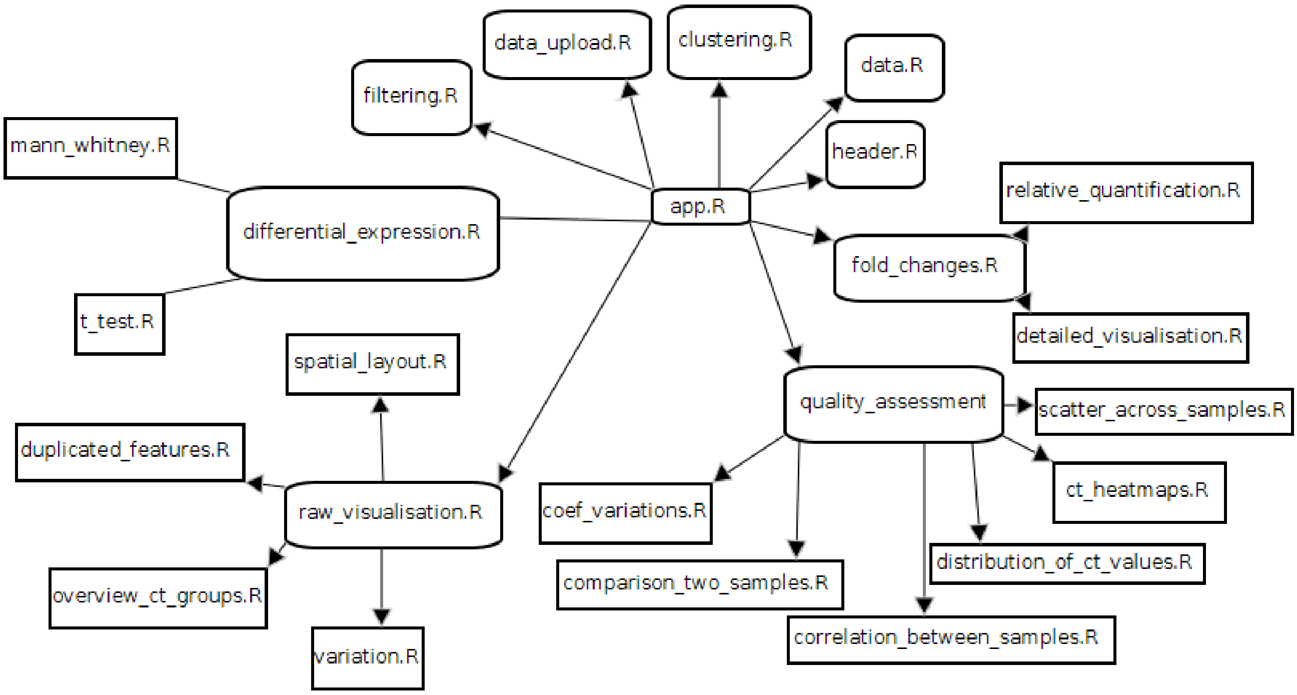

FIGURE 4.20: Application file structure

The web application is highly modularised. There is an app.R file, which the web application is started from and it calls to different modules that are found in the modules folder. Some modules call submodules because in other circumstances, they could have been complicated to develop.

4.9.2 Executing web application

In order to execute the application, there are some installation requirements:

- R version 3.2.3
- Shiny 0.14 (R web framework) (Chang et al., 2016)
- HTqPCR 1.24.0 from Bioconductor
- DT 0.2 (for special datatables)

It is possible to clone the repository from https://github.com/nachoyellow/qPCRgui (Available since 02-Nov)

After that, it is enough to execute this command from a R console: shiny::runApp() where app.R is located. This will automatically load any required packages.

The web application has been deployed in shinyapps.io and it is possible to access it from here: https://nacho-projects.shinyapps.io/qPCRgui/ (available since 03-Oct)

4.9.3 qPCR data and files' structure

In this section, qPCR data internal structure and how it could be imported from files will be introduced. Although HTqPCR package offers a variety of functions to handle different input data, the web application is quite limited in that aspect and it only admits a specific file structure that will be explained below. That is because it would require too much time to handle different inputs and that is out of the scope of this project.

Internally, qPCR data is stored in an object class called qPCRset which inherits from eSet. eSet class is designed to handle data where the same property is measured across a range of samples. qPCR data measures Ct of genes across a range of samples, which fulfils the requirements.

qPCRset object can contain the following information:

- Feature names: a list of strings containing the names of the features (i.e., DNA samples=genes).
- Sample names: a list of strings containing the names of the samples (i.e., people, cats, mice)
- Expression matrix: a matrix containing the Ct values for each sample-feature combination.
- Flag: a table containing an indicator for each Ct value as e.g. "Passed", "Flagged", "Failed. Provided by input file.
- Feature type: a list of strings containing the different types of features.
- Feature position: a list of strings containing the location of a gene in a defined spatial layout.
- Feature category: a table representing the quality of the measurement for each Ct value, e.g. "OK", "Undetermined", "Unreliable". Set by the application.

The web application gives access to the feature and sample names, the type and position of the features and expression matrix. It also gives the chance to delete features and samples. However, it is not possible to modify them after they are load or get them back without reloading raw data again, any modification on those must be done directly in the input files. This functionality is not part of the scope of the project, because it is more software-oriented and it is not so essential to analyse data.

Files' structure: input data format

One of the expected files of the application is **files.txt**, a tab-delimited meta data file which must have in the first column the name of the sample files. Its header must be: **File**. It's necessary to have an additional column which represent groups of samples. More than one group column is allowed. For example:

```
File Treatment
sample1.txt Control
sample2.txt LongStarve
sample3.txt LongStarve
sample4.txt Control
sample5.txt Starve
sample6.txt Starve
```

Now, every sample file defined by files.txt needs to be provided with it. They are expected to be in standard "plain" format, which is a tab-delimited file containing no header. Make sure that the feature position is in the 3rd column, flag is in the 4th , the feature name is in the 6th , the feature type is in the 7th , the Ct value is in the 8th.
For example, sample1.txt file:

```
1 Control A1 Passed Sample01 Gene1 Endogenous Control 11.463166 FALSE
2 Control A2 Passed Sample01 Gene2 Target 33.949196 22.479778 0.26758063 0
1 0.4257367 2.3488696 FALSE
3 Control A3 Passed Sample01 Gene3 Target 27.956657 16.195972 0.14037517 0
1 0.63891655 1.5651497 FALSE
```

It has more information, but essentially it fulfils the requirements because feature positions ["A1", "A2", "A3"] are in the 3rd column, flags ["Passed", "Passed", "Passed"] are in the 4th, feature names ["Gene1", "Gene2", "Gene3"] are in the 6th , feature types ["Endogenous Control", "Target", "Target"] are in the 7th, and Ct values [11.46, 33.95, 27.95] are in the 8th.

Chapter 5

Conclusions and project further improvements

The aim of this chapter is to state conclusions of the whole project as well as possible further improvements that could be developed within it.

First, it needs to be mention that this project demanded to dive right in a completely new field (bioinformatics) for the student. Its essential concepts related to biology and statistics are also introduced. The project also demanded to become familiar with a new technology for creating web applications (Shiny). Since the bioinformatics field was completely new and seemed so theoretical for the student, I decided to get in touch with the researchers in Leioa to have a feeling of the state of the art in the field and know their everyday practice needs related to qPCR data analysis. I can conclude that I have taken advantage from this chance to actually understand the field from a more practical perspective.

When it comes to analyse the results of the project, the expectations and objectives of the project have been way over accomplished. However, the improvements that could be developed to the project are numerous. In the following paragraphs, the most relevant ones will be mentioned.

In the scope of qPCR data analysis, different normalisation methods have been introduced. They may be further developed. When it comes to statistical testing, a deeper research could have been performed together with a well-explained background and examples.

In the scope of the web application development, more features could be implemented with their corresponding specifications and explanations such as new filtering methods, new improved and more meaningful interactive plots, or more complex statistical tests. Moreover, error handling and a better overall visualisation could be accomplished from a more software-oriented perspective. In this project, the focus has been developing the web in order to analyse qPCR data, so simplicity has been chosen with respect to the development phase.

Appendix A

Detailed time estimation

TABLE A.1: Detailed time estimation

Tasks	# Hours
Fundamental research	**20**
Fundamental biological concepts	5
Fundamental descriptive statistics and inference	15
Management	**10**
Web design	**10**
Task planning	2
Draw screens	8
HTqPCR research and web development	**200**
Load expected data	**15**
Research	5
Web implementation	10
Raw data visualisation	**20**
Research	10
Web implementation	10
Filtering	**20**
Research	10
Web implementation	10
Normalisation	**25**
Research	15
Web implementation	10
Quality assessment	**25**
Research	15
Web implementation	10
Data clustering	**25**
Research	15
Web implementation	10
Differential expression	**35**
Research	25
Web implementation	10
Fold changes	**35**
Research	25
Web implementation	10
Dissertation design	**5**
Dissertation development	**55**
Total number of hours	**300**

Bibliography

Bustin, Stephen A. (2012). *Definitive qPCR: Basic Principles*. Stephen A. Bustin.

Chang, Winston et al. (2016). *shiny: Web Application Framework for R*. R package version 0.14. URL: https://CRAN.R-project.org/package=shiny.

Crawley, Michael J. (2007). *The R Book*. Wiley John + Sons.

Dvinge, Heidi and Paul Bertone (2009). "HTqPCR: High - throughput analysis and visualization of quantitative real - time PCR data in R". In: *Bioinformatics* 25(24), p. 3325.

Ewens, Warren J. and Gregory R. Grant (2005). *Statistical Methods in Bioinformatics: An Introduction*. 2nd ed. Statistics for Biology and Health. Springer-Verlag New York.

Hahne, Florian et al. (2008). *Bioconductor Case Studies*. 1st ed. Springer Publishing Company, Incorporated. ISBN: 0387772391, 9780387772394.

Nelson, David L. and Michael M. Cox (2012). *Lehninger Principles of Biochemistry*. W.H. Freeman.

Tan, P., M. Steinbach, and V. Kumar (2005). *Introduction to Data Mining*. Pearson.