REPLACING QPCR NON-DETECTS WITH MICROARRAY EXPRESSION DATA

An initialized approach towards microarray and qPCR data integration

Master Degree Project in bioinformatics
One year level
30 hp
Spring term 2018

Author: Jonas Sehlstedt
a13jonse@student.his.se
Bioinformatics

Supervisor: Angelica Lindlöf
angelica.lindlof@his.se

Examiner: Björn Olsson
Bjorn.olsson@his.se

School of Bioscience
University of Skövde
Box 408
54128 Skövde, Sweden
Abstract

Gene expression analysis can be performed by a number of methods. One of the most common methods is using relative qPCR to assess the relative expression of a determined set of genes compared to a reference gene. Analysis methods benefit from an as homogeneous sample set as possible, as great variety in original sample disease status, quality, type, or distribution may yield an uneven base expression between replicates. Additionally, normalization of qPCR data will not work if there are missing values in the data. There are methods for handling non-detects (i.e. missing values) in the data, where most of them are only recommended to use when there is a single, or very few, value missing. By integrating microarray expression data with qPCR data, the data quality could be improved on, eradicating the need to redo an entire experiment when too much data is missing or sample data too is heterogeneous. In this project, publicly available microarray data, with similar sample status of a given qPCR dataset, was downloaded and processed. The qPCR dataset included 51 genes, where a set of four DLG genes has been chosen for in-depth analysis. For handling missing values, mean imputation and inserting Cq value 40 were used, as well as a novel method initialized where microarray data was used to replace missing values. In summary replacing missing values with microarray data did not show any significant difference to the other two methods in three of the four DLG genes. From this project, it is also suggested an initialized approach towards testing the possibility of qPCR and microarray data integration.
**List of Abbreviations**

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ANOVA</td>
<td>Analysis of variance</td>
</tr>
<tr>
<td>APC</td>
<td>Adenomatous Polyposis Coli (gene)</td>
</tr>
<tr>
<td>BMI</td>
<td>Body Mass Index</td>
</tr>
<tr>
<td>CI</td>
<td>Confidence interval</td>
</tr>
<tr>
<td>Cq</td>
<td>Cycle threshold (commonly abbreviated ‘Ct’)</td>
</tr>
<tr>
<td>CRAN</td>
<td>Comprehensive R Archive Network</td>
</tr>
<tr>
<td>DEG</td>
<td>Differentially Expressed Gene</td>
</tr>
<tr>
<td>DLG</td>
<td>Disks large (gene)</td>
</tr>
<tr>
<td>EM</td>
<td>Expectation-maximization</td>
</tr>
<tr>
<td>GAPDH</td>
<td>Glyceraldehyde-3-Phosphate Dehydrogenase (Gene)</td>
</tr>
<tr>
<td>GEO</td>
<td>Gene Expression Omnibus</td>
</tr>
<tr>
<td>HGNC</td>
<td>HUGO Genome Nomenclature Committee</td>
</tr>
<tr>
<td>HPRT1</td>
<td>Hypoxanthine Phosphoribosyltransferase 1 (Gene)</td>
</tr>
<tr>
<td>IBS</td>
<td>Irritable Bowels Syndrome</td>
</tr>
<tr>
<td>IDA</td>
<td>Iron Deficiency Anemia</td>
</tr>
<tr>
<td>IHC</td>
<td>Immunohistochemistry</td>
</tr>
<tr>
<td>IPO8</td>
<td>Importin 8 (Gene)</td>
</tr>
<tr>
<td>k-NN</td>
<td>k-Nearest Neighbor</td>
</tr>
<tr>
<td>LIN7</td>
<td>Protein lin-7 (gene)</td>
</tr>
<tr>
<td>miRNA</td>
<td>MicroRNA</td>
</tr>
<tr>
<td>MIQE</td>
<td>The Minimum Information for Publication of Quantitative Real-Time PCR Experiments</td>
</tr>
<tr>
<td>MRPL</td>
<td>Mitochondrial Ribosomal Protein L (Gene)</td>
</tr>
<tr>
<td>NA</td>
<td>Not Applicable</td>
</tr>
<tr>
<td>NGS</td>
<td>Next Generation Sequencing</td>
</tr>
<tr>
<td>PCR</td>
<td>Polymerase Chain Reaction</td>
</tr>
<tr>
<td>PPIA</td>
<td>Peptidylprolyl Isomerase A (Gene)</td>
</tr>
<tr>
<td>PTEN</td>
<td>Phosphate and tensin</td>
</tr>
<tr>
<td>qPCR</td>
<td>Quantitative Polymerase Chain Reaction</td>
</tr>
<tr>
<td>RF</td>
<td>Random Forests</td>
</tr>
<tr>
<td><strong>RPLP0</strong></td>
<td>Ribosomal Protein Laternal Stalk Subunit P0 (Gene)</td>
</tr>
<tr>
<td><strong>SD</strong></td>
<td>Standard deviation</td>
</tr>
<tr>
<td><strong>SF3A1</strong></td>
<td>Splicing Factor 3a Subunit 1 (Gene)</td>
</tr>
<tr>
<td><strong>SVM</strong></td>
<td>Support Vector Machine</td>
</tr>
</tbody>
</table>
# Table of Contents

Abstract ................................................................................................................................. 1
List of Abbreviations ........................................................................................................... 2
Table of Contents ............................................................................................................... 4
Introduction ............................................................................................................................ 1
  Colorectal Cancer .............................................................................................................. 1
  Biomarkers and Differentially Expressed Genes (DEG’s) .................................................. 1
  Microarray and Quantitative Polymerase Chain Reaction (qPCR) Data ............................ 2
Data Analysis ......................................................................................................................... 3
Problem Definition and Motivation ..................................................................................... 3
  Sample Heterogeneity ......................................................................................................... 3
  Missing Values .................................................................................................................... 4
  DLG and LIN7 Genes ......................................................................................................... 4
  Pipelines and Data Integration .......................................................................................... 4
  Aim .................................................................................................................................... 5
Materials and Methods .......................................................................................................... 5
  The R Language .................................................................................................................. 5
  RT-qPCR ............................................................................................................................. 5
  Microarray .......................................................................................................................... 6
  Reference Genes ............................................................................................................... 6
  Integrating Data .................................................................................................................. 6
  Relevant methods not used ................................................................................................. 7
Results .................................................................................................................................. 8
  qPCR data .......................................................................................................................... 8
  Microarray data .................................................................................................................. 11
  Normalization and Combining Data ................................................................................... 12
Discussion ............................................................................................................................... 18
  qPCR data quality control ................................................................................................. 18
  Microarray data collection ................................................................................................. 18
  Handling NAs and combining data .................................................................................... 18
Introduction

Colorectal Cancer

Cancer is a publically well-known disease, and it is the second leading cause of death globally. About 1/6 of deaths are due to cancer, and in 2015 alone 8.8 million people world-wide died from some type of cancer (WHO | Cancer, 2017). In Europe the second most common cause of cancer death in women is colorectal cancer (after breast cancer), being the third most common cause of cancer death in men (following lung and prostate cancer). Approximately 60% of all new cases of colorectal cancer occur in countries classified as high-income (Colorectal cancer, 2017). In Sweden, deaths from malignant tumors in the colon or rectum are around 2500 cases annually, with that number relatively unchanged since 1997 (Statistikdatabas för dödsorsaker, 2017). There is no distinct difference between colorectal cancer and colon cancer, and the names are often used inter-changeably.

Biomarkers and Differentially Expressed Genes (DEG’s)

There are several fields of medical research focusing on cancer prevention, detection, treatment, and so forth. The majority of the relevant fields rely on experimental methods where, for example, cancer and control cells are treated with a potential drug, and hopefully decrease in cancer cell amount/volume/mobility (etc.) is compared to, optimally, an unaltered state of the healthy control cells. In bioinformatics a common goal for cancer research is to find useful biomarkers in large scale data sets (Crichton et al., 2010). A biomarker, a linguistic blend of “biological marker”, is in general referred to as an indicator of some sort of biological state or condition that can accurately and in repeated cases be measured. This can be blood pressure (Desai et al., 2006), presence of certain antibodies (Sirotti et al., 2017), changes in vascular structures (Ravi Teja et al., 2017), gene expression patterns (Rázga and Némethová, 2017), and more. The term ‘biomarker’ was first used in the 1950’s (Porter, 1957), but was not generally used until around the 80’s (Paone et al., 1980) (Aronson, 2005). In 1998 the National Institutes of Health Biomarkers Definitions Working Group defined the term ‘biomarker’ as a “characteristic that is objectively measured and evaluated as an indicator of normal biological processes, pathological processes, or pharmacological responses to a therapeutic intervention” (Strimbu and Tavel, 2010). The greatest advantages of using biomarkers as so called ‘surrogate endpoints’ (Boone and Kelhoff, 1993) are that they are much easier and cheaper to measure in comparison to ‘true’ endpoints; it is much easier to screen for a certain auto-antibody in a blood sample than to wait for a patient to display all physical, and maybe painful, symptoms of an autoimmune disease. A biomarker also allows for earlier measurement (Aronson, 2005). There are general guidelines for criteria that, if fulfilled, indicate if a biomarker is a good surrogate endpoint or not (Table 1). These guidelines are derived from Austin Bradford Hill’s guidelines that increase the likelihood that an association is causative, originally proposed in the context of environmental factors in disease development (Hill, 1965) (Aronson, 2005).
Table 1. The eight Austin Bradford Hill’s guideline terms with short descriptions of each term.

<table>
<thead>
<tr>
<th>GUIDELINE TERMS</th>
<th>DESCRIPTION OF CHARACTERISTIC</th>
</tr>
</thead>
<tbody>
<tr>
<td>ANALOGY</td>
<td>There are similar results to which a relationship can be associated to</td>
</tr>
<tr>
<td>BIOLOGICAL</td>
<td>Increasing exposure to an intervention produces increasing effects on the marker and the disease</td>
</tr>
<tr>
<td>GRADIENT*</td>
<td></td>
</tr>
<tr>
<td>COHERENCE</td>
<td>Association is consistent with the natural history of the disease and marker</td>
</tr>
<tr>
<td>CONSISTENCY</td>
<td>Association persists in different individuals, places, circumstances, and times</td>
</tr>
<tr>
<td>EXPERIMENTAL</td>
<td>An intervention gives results consistent with the association</td>
</tr>
<tr>
<td>EVIDENCE</td>
<td></td>
</tr>
<tr>
<td>PLAUSIBILITY</td>
<td>Credible mechanisms connect the marker, pathogenesis of the disease, and the mode of action of the intervention</td>
</tr>
<tr>
<td>SPECIFICITY</td>
<td>Marker is associated with specific disease</td>
</tr>
<tr>
<td>STRENGTH</td>
<td>There is a strong association between marker and outcome (or between effects of treatment on each of the two)</td>
</tr>
</tbody>
</table>

* Dose-responsiveness

Identifying sets of differentially expressed genes (DEG’s) in a homogeneous group of test samples compared to control samples has been used as a method for finding biomarkers repeated times in previous studies (Ge et al., 2018) (Hibbs et al., 2004) (Cho et al., 2009). A set of DEG’s can also be referred to an ‘expression pattern’ of either down- or up-regulated genes seen in a specific physiological or pathophysiological state of a given sample. If the same expression pattern is seen in repeated sets of, for example, colorectal cancer tumors in a certain phase of cancer development, an unlabeled sample can be screened to determine that it is in a colorectal cancer state given the presence of the set pattern (Wang et al., 2015) (Castillo et al., 2017).

**Microarray and Quantitative Polymerase Chain Reaction (qPCR) Data**

In the late 1980’s/early 1990’s a method for producing observable amounts of specific DNA or RNA sequences of any given length was developed. This method, known as polymerase chain reaction (PCR), uses oligonucleotide sequences called ‘primers’ matching the desired DNA or RNA sequence, and generates multiple copies of the fragment, while the rest of the genome is copied in very small quantities (Medicine, 2018). In later updates of the method ways to accurately measure the relative (or absolute) expression levels of individual genes has been developed. In quantitative PCR (qPCR) expression levels relative to one another may be measured. At a certain PCR cycle amplification of a desired gene, the expression signal will exceed disruptive background noise, thus making expression detectable. The cycle at which expression is seen is called the cycle threshold (Cq). The Cq is compared to other Cq values of other genes, one or more control genes for example, giving expression values relative to each other (Livak and Schmittgen, 2001).

Expression analysis using qPCR methods require a predetermined, and often relatively small, set of genes of interest. Using so called ‘microarray’ technologies opens up for the possibility to measure expression of sets of tens of thousands of genes. From sensitive mRNA complimentary DNA (cDNA) is synthesized, which is much more stable than the former. These cDNA samples are loaded onto microarray chips. The chips are full of microscopic wells containing probes corresponding to different genes. The DNA samples are fragmented, and to each fragment a fluorescent marker is added, and
when bound to matching probes on the chip, fluorescent emission is measured to estimate gene expression levels (Govindarajan et al., 2012).

Microarray and qPCR methods are not the only ways to generate expression level data, in the last decade more robust protocols for next generation sequencing (NGS) has been developed. NGS does foremost provide possibilities and approaches for full- and part-of-genome studies (Behjati and Tarpey, 2013). High-throughput gene expression analysis can be performed using RNA-Sequencing and NGS technologies. The field of NGS technologies is however still limited by relatively novel and not so robust analysis protocols as well as requiring possibly large computer capacities, where easier and well developed methods are available for microarrays and PCR technologies (Hardwick et al., 2017) (Jain, 2017).

Data Analysis

There are several robust analytical methods developed for both qPCR and microarray data. DNA microarray experiments can yield several thousands of gene expression values where qPCR more often focuses on the expression of a smaller, hand-picked group of genes. For microarray data there exist both web-based (Xia et al., 2005) and computational (Pelizzola et al., 2006) (Du et al., 2017) (Ritchie et al., 2015) tools, such as specific analytical packages for R language (Ritchie et al., 2015) (Gautier et al., 2017). Analytical tools are often used in microarrays to find co-regulation of genes or groups of genes, as well as finding and building networks of interacting genes and genetic components (Cho et al., 2009) (Wang et al., 2013) (Stäehler et al., 2012). Microarray data is often analyzed with the purpose of finding gene expression patterns in a large gene set, with very limited predetermined insight in which specific genes to look at. Often machine learning (Greene et al., 2014) is used to find the most statistically significantly differentially expressed genes in an entire gene dataset. With qPCR it is the opposite of microarray data; a predetermined set of genes which have shown to be potential significant candidates for the research at hand is analyzed. The most common method of analyzing (relative) gene expression in qPCR data is the $2^{\Delta\Delta C_T}$ method (Livak and Schmittgen, 2001). In this method the relative expression of the samples are normalized, usually to preset control probes known to have even expression between biological and technical replicates. The normalized values are translated to fold change values, which show how many times more (or less) a gene is “expressed” than another gene.

In gene expression profiling pipelines are often developed to allow for easier repetition with, for example, alternative datasets addressing the same issue or subject. A microarray pipeline is typically initiated by inputting raw array data, and methods specific for the type of data in question developed for quality assessment, outlier removal, normalization, etc. (Castillo et al., 2017) The end product of a pipeline of this type is the gene expression levels. When it comes to qPCR data analysis it mostly relies on robust, well established manual methods. Due to the relatively small volume of produced data a high-throughput bioinformatic approach is rarely needed. Data analysis is in general performed by the same researchers who performed the experiment (wet lab). However lately more general analysis models and pipelines have been developed to include bioinformatics in qPCR analysis (Pabinger et al., 2014).

Problem Definition and Motivation

Sample Heterogeneity

For a PCR analysis of gene expression the genes of interest can be compared between healthy and sick patients. Samples obtained through biopsies are rarely taken from completely healthy patients, and one or more secondary disorders are often the basis for allowing a biopsy to be performed. For example, in the given qPCR data set test sample biopsies are taken from patients suffering from
colorectal cancer and control sample biopsies taken from patients not in a cancerous state but who have some other disorder allowing for biopsies. These disorders include IDA, IBS, rectal bleeding, and more. All of these disorders may by their selves show alternate genetic signatures compared to what would be seen in a healthy individual. This might be good for investigating and developing methods easily differentiating colon cancer patients from patients suffering from other colorectal disorders. However, this is not a very robust way to ensure the role of a gene of interest in a certain physiological state. If the control sample data could be “homogenized” by the use of publically available microarray data, maybe more robust conclusions can be made from the obtained patient sample data, lowering the time and resources wasted towards a rejected hypothesis due to insufficient stability in the samples.

**Missing Values**

Another problem that can occur in qPCR data analysis is the occurrence of lacking data points and values. This may be because of experimental errors as well as a gene being expressed at a level low enough that it is considered a non-detect (McCall et al., 2014). There have been several methods developed towards addressing these problems, and statistical analysis is still possible despite missing values (McCall et al., 2014) (Sherina et al., 2017).

**DLG and LIN7 Genes**

The DLG gene produces a scaffolding protein which recruits channels, receptor, and signaling molecules to the plasma membrane in polarized cells (UniProt, 2018). There exists five orthologues for the gene; DLG-1, DLG-2, DLG-3, DLG-4, and DLG-5. Some of these homologues, such as DLG-1 and DLG-3, have shown tumor suppressive abilities, and DLG-1 has been shown to interact with APC and PTEN in G0/G1-to-S phase progression (Lin et al., 2015). The LIN7 gene expresses a similar product, a protein also included in the distribution and maintenance of receptors and channels at the membrane of polarized cells (UniProt, 2017). There are three homologues of the LIN7 gene; LIN7a, LIN7B, and LIN7c.

In general there seem to be common for amplification and upregulation of polarity genes in cancer development (Lin et al., 2015). Previous research has described the interaction between DLG-2 and LIN7 and suggested that there is a correlation between the co-regulated expression of the genes and tumor development. Specifically there seems to be a correlation between the upregulation of these genes and the progression of neuroblastoma (Personal communication, Simon Keane, 2018.02.12). This has however not been investigated in colorectal cancer and significant expression changes in the genes may serve as a biomarker for the disease.

**Pipelines and Data Integration**

Previous work has been done to integrate different types of data into one combined pipeline, such as microarray and RNA-seq. data (Castillo et al., 2017). There has previously been no work published on integrating relative qPCR data with high through-put data such as microarray data. This might be because of the two methods being used separately in different experiments, or one method may be preferred over the other. Microarrays and qPCR are in general used for different analyses; microarrays are used when the target genes are unknown and qPCR when target genes are predetermined. The greatest obstacle in integration of these types of data is the relative nature of the qPCR data. This has to be taken into consideration and a method found where the relative data may be comparable to the more static microarray data in a robust way. Microarray data also exists in a more dynamic spectrum in the form of time-series gene expression data (Androulakis et al., 2007), which may be a more suitable comparison to the relative qPCR data, and may be analyzed if the more static alternative proves to be incompatible. Previous research has studied the compatibility between microarray and qPCR data. While the research points to complementarity of the two data types, it also highlights the
Aim

This research aims to improve the quality of heterogeneous qPCR sample data by integrating it with more homogeneous microarray data. By using publically available microarray data for normalizing the given qPCR data set a more robust conclusion can be drawn from the potential DEG’s and expression signatures seen in the data. This will also improve on the limited control sample size arising from when sample biopsies are difficult to obtain from healthy subjects. There is as well a biological contribution to cancer research if the DLG and LIN7 genes are shown to be significantly regulated in colorectal cancer.

Materials and Methods

The R Language

R is a programming language well suited for statistical analysis and graphical visualization. R provides several statistical methods, such as linear and non-linear modeling, clustering, classical statistical tests, as well as the possibility to generate publication-quality plots (Team, 2011). R is easily expanded by downloading and installing packages which can be tailored to a specific task. Packages are available at the public repositories CRAN (Maintainers, 2018) and Bioconductor (Bioconductor, 2018).

Where R differs from other programming languages, such as JAVA and Python, is that it was designed specifically for in-depth statistical analysis and high-quality reporting, sacrificing performance and some of the flexibility which comes with a more object oriented language.

RT-qPCR

Using real-time quantitative reverse transcription PCR (qPCR) the relative level of expression in different genes may be measured. Instead of generating an estimate of the total amount of gene expression like a standard PCR a qPCR allows for detection and measurement of products generated each PCR cycle (Heid et al., 1996) (Real-Time qRT-PCR, 2018). The most common method of analyzing (relative) gene expression in qPCR data is the $2^{ΔΔC_T}$ method (Livak and Schmittgen, 2001). In this method the relative expression of the samples are normalized, usually to preset control probes known to have even expression between biological and technical replicates. The normalized values are translated to fold change values, which show how many times more (or less) a gene is “expressed” than another gene.

The qPCR data for this project was granted by a research group at the University of Skövde. The data set is a tab-delimited .xlsx file containing the expression values as well as clinical data on the sample patients. Raw Cq-values as well as fold change values are present. The fold change of gene expression data is presented in a binary logarithm ($\log_2$) scale. Clinical data include age and sex of the patient, as well as tumor stage and localization. It also includes reason for colon biopsy performed in each control subject.

Regarding qPCR data analysis, this step mainly relies on robust, well-established manual methods. Due to the relatively small volume of produced data a high-throughput bioinformatic approach is rarely needed. Data analysis is in general performed by the same researchers who performed the experiment (wet lab). However lately more general analysis models and pipelines have been developed to include bioinformatics in qPCR analysis (Pabinger et al., 2014).
There are mainly two qPCR analysis R-packages that will be used for this project. The HTqPCR package can be used for quality assessment, normalization, visualization, and statistical significance testing in Cq values between different features, such as genes (Dvinge and Bertone, 2018). This package is however not fully compliant with the MIQE-guidelines (Bustin et al., 2009), which the qpcR package is (Spiess, 2015). The latter package includes a larger library of qPCR data analysis methods than the former, and can be used for raw data handling such as Cq value calculation. The qpcR package is not as user friendly as the HTqPCR package and does not feature the same graphical interface and visualization, and it does not include methods for statistical analysis (Pabinger et al., 2014).

**Microarray**

Microarray data for this project is downloaded and collected from the online NCBI GEO database. Basic criteria for which datasets are viable for this research are based on former colorectal cancer gene expression analysis (Shangkuan et al., 2017). Criteria are added to further narrow down the search results. Currently a search for the term “colon cancer” yields 7376 hits. Filtering search by only showing results of entry type “series”, organism “human”, study type “expression profiling by array”, as well as attribute name “tissue” narrows down the results to 79 hits. Further filtering out only datasets including .CEL supplement files yields 41 hits, all including experiments using the AffyMatrix microarray platform.

All the microarray datasets included in this project utilize the Affymetrix Human Genome U133 Plus 2.0 Array [HG-U133_Plus_2] which contains roughly 54,000 probe sets, used to analyze 47,000 transcripts including almost 39,000 human genes (Affymetrix, 2003). The datasets can be downloaded and imported to the R software using the GEOquery package (Davis, 2018). There are several packages developed for quality assessment, preprocessing, and expression analysis. This research will focus on using three of the most common packages; limma (Smyth et al., 2018), siggenes (Schwender, 2018), and affy (Iririzarry, 2018). The specific genes presented in the RT-qPCR dataset can be identified in the downloaded microarray datasets using the biomaRt (Durnick et al., 2018b) packages.

**Reference Genes**

Included in the qPCR data set is the reference gene GAPDH, which has been considered a stable and suitable gene for gene expression analysis in colorectal cancer studies as well as in other disease states (Zainuddin et al., 2010) (Kozera and Rapacz, 2013) (Krzystek-Korpacka et al., 2016). At the same time several studies have highlighted the need to evaluate potential reference gene for each individual experiment performed, instead on simply relying on using the same reference (GAPDH), simply because it has previously shown stability (Sørby et al., 2010) (Kozera and Rapacz, 2013, Aithal and Rajeswari, 2015). Other genes and gene pairs have been suggested as reference genes in expression level studies associated with colorectal cancer in some way; PPIA, SF3A1, MRPL, RPLPO, IPO8 (Aggerholm-Pedersen et al., 2014) (Krzystek-Korpacka et al., 2016), and gene pairs HPRT1/PPIA and IPO8/PPIA (Sørby et al., 2010). The varying suggestions for suitable ways of dealing with reference genes does in itself suggest that different approaches (or genes) should be tested during implementation.

**Integrating Data**

There will be three main approaches towards increasing data quality in this research.

The first approach will be to test a direct integration of the two data types using already developed integration methods for microarray and NGS integration, replacing NGS data with qPCR data. The integration will be based on the model developed in *Integration of RNA-Seq data with heterogeneous microarray data for breast cancer profiling* (Castillo et al., 2017). In this model microarray and RNA-
Seq data are individually processed to obtain the expression values of each dataset. Sample data from both microarray and RNA-Seq technologies are integrated using the merge function from the R base package. Normalization is then carried out using the normalizeBetweenArrays function from the limma package. Integrated gene expression levels and DEG extraction at both individual and integrated levels are followed by classification using support vector machines (SVM) (Noble, 2006), random forest (RF) (Díaz-Uriarte and Alvarez de, 2006), and k-nearest neighbor (k-NN) (Parry et al., 2010) technologies. Following these steps, overlapping DEG’s, found in both the individual datasets as well as the integrated model are considered viable biomarker candidates. Testing this integration model, replacing the RNA-Seq data with qPCR data, as well as narrowing down the microarray data to only include genes from the qPCR data set will give an initial indication on the quality, compatibility, and relative expression of the targeted genes.

Secondly, missing qPCR data values will be replaced with microarray data. This method will be compared to using pre-developed packages for non-detects, such as nondetects (McCall and Sherina, 2018).

Finally it will be tested if data quality is improved by integrating or replacing the qPCR control data with microarray data. The HtqPCR R package (Dvinge and Bertone, 2018) takes tab-delimited text files containing Cq values and feature identifiers (such as gene ID’s) as input. In this research, the qPCR data is available in .xlsx file format, which is very simple to convert to a suitable .txt file if necessary. There are several functions for data visualization in the R package, such as displaying Cq and fold change values in bar plots of genes in one or more samples. Density distribution, box plots, scatterplots, and histograms can be generated for data quality control. HtqPCR does allow for normalization using the \(2^{-\Delta\Delta CT}\) method (Livak and Schmittgen, 2001) as well as more refined methods (Dvinge and Bertone, 2009). Finally statistical analysis can be performed using t-test between two conditions. The instructions also refer to using the limma R package for more statistical analysis options (Dvinge and Bertone, 2009) (Smyth et al., 2018).

Exactly how to implement microarray data into the non-detects and qPCR control data is going to be evaluated “as it comes up” in the implementation. There have been very little previous work done on the integration of qPCR and microarray data, and there are quite many different packages for R being used for this research. This makes it very difficult to anticipate all the obstacles and errors that can show up.

**Relevant methods not used**

There are several previous models developed for combination and integration of microarray and NGS data (Chavan et al., 2013) (Bauer et al., 2014) (Lyu and Li, 2016) (Swindell et al., 2014). The Mayday SeaSight is an extension of the previously established Mayday platform, which is at its core a tool for visualization, analysis, and storage of microarray data (Dietzsch et al., 2006). The SeaSight extension allows for integration of microarray and deep sequencing as well as combining data from other platforms (Battke, 2011). The MayDay platform is Java based, and while it could be used for this research it is more depending on programming, and additional plug-ins are needed for connecting it to the R language software. Though the platform might be a relevant approach to the project it is an added process to make sure that it is properly integrated to the R software. This adds one more aspect to a project already heavy on method development, and to minimize the potential cross-function errors the model development should be kept to the same platform as much as possible; in this case the R language.

Missing values are often the result of weak expression signal. The signal, even if present, is so small that the PCR reaction reaches its final cycle before the signal can be measured. Some researchers simply replace the empty data points with a Cq value equal to the maximum PCR cycle number, which
typically equals to 40. This has however shown to introduce great bias, and basing the analysis on the nondetects package is more robust (McCall et al., 2014).

Results

qPCR data

The qPCR data was provided by a research group at the University of Skövde in the form of an .xlsx file, with expression data of the different genes and samples divided into several sheets. The file also included clinical data for the patients on which the tissue biopsies were performed. This is the only original source of qPCR data used throughout the whole implementation.

As a first step the data had to be organized to fit the first intended processing method in R which is the “HTqPCR” package (Dvinge and Bertone, 2018). This package accepts tab delimited .txt files with sample name, expression values, and other sample information organized into a set order of columns (specified by the “readCtData” function) to generate a qPCRset object. This is an object similar to the eSet used for handling microarray data (Gentleman et al., 2018) adapted for qPCR data. The original data is structured with each tissue sample (here from referred to as just sample ‘sample’) categorized into which patient it was obtained from, with each column indicating the Cq-value for each gene tested. Controls (samples obtained from patient without colorectal tumors) and targets (colorectal tumor patients) are listed in separate sheets in the document. The samples were transferred to another .xlsx file where the samples were distributed over 159 sheets, each sheet including a single sample. The 159 samples were made up of 64 controls and 95 targets, not taking the different tissues (from where the samples were obtained) into account. The columns of the sheets included information on (from left to right) row number, original tissue type, position of the sample on the qPCR plate, flag status, patient number, HGNC gene symbol, gene status (target or control), expression, and lastly the health status of the patient. The order the information is listed is decided by the “readCtData” function as it demands for this order to properly build a qPCRset. That is why the position of each sample is included as “NA” instead of removed completely, due to there not being any data available for the plate layout and position of each individual gene and sample. The original experiment was performed in triplicates; meaning that each gene yielded three Cq values for each sample. However the Cq values available in the given dataset are the average values of these triplicates.

Additionally, a .txt file was created listing the sample names and control/target status. This was simply used to refer to the samples when importing them into R (appendix A).

The expression matrix from the qPCRset was put into a separate object and the row names set to match the genes (appendix A), making referencing the expression a little bit easier as well as observing the data more convenient. As a first visualization a bar plot was generated, showing the average Cq values of the first couple of listed genes (figure 1). Bars are missing for either control and/or target samples, due to NA’s being present in the at least one sample in either the control or target (or both) groups.
Figure 1. Bar plot of first sixteen genes. Y-axis show average Cq values for each gene. X-axis list the gene names. Red bars indicate control samples, and green bars target samples. Blue rectangle is used to highlight the four DLG genes which will be returned to later in the analysis.

Studying the heatmap (figure 2) it is possible to see that there are several genes for which at least one sample is missing expression data. The relationships between genes and samples seems to be only based on patterns of present NAs. Even though the plot does not show sample names it gives an overview of how much missing data that is present in the data. The heatmap was created from the \texttt{qPCRset}, using the HTqPCR function “plotCtCategory”. The sample names have been excluded from the heatmap because the text format makes the individual names indefinable. The figure does include both control and target samples, but the grouping is arbitrary for an overview of the NA’s present in the whole dataset. It would be beneficial for the analysis to have heatmaps generated for each of the NA handling methods, visualizing the differences in relationships between genes and samples changing before and after processing. This was however not possible due to the HTqPCR package not accepting eSets as inputs for the function, only qPCRsets, despite them being similar in structure. Due to the multiple NAs further analysis of the qPCR data is not possible in its current state. A common approach to handling NAs in qPCR data is using the “nondetects” package, which uses an expectation-maximization (EM) algorithm to predict which values are most likely to fit into the NA points (McCall and Sherina, 2018) (McCall, 2014). However, the package did not work on this data set, as trying to initialize the method yielded only error messages (appendix A), that could not be solved. Additionally, the package could not start the function unless sample names in the \texttt{qPCRset} matched the column names in the expression matrix in the same \texttt{qPCRset}, which it actually did but R did not recognize (Appendix A)
Instead of “nondetects” the method “mean imputation” as well as replacing NA’s with Cq 40 values was used. For this, a new Excel-sheet was created with a structure similar to that of the expression data matrix in an eSet; with samples listed on the top row, genes in the first column, and with matching Cq values filling the cells. The samples were also marked with either “control” or “target” status. First, for mean imputation, the average of the present values in each gene and status group (control/target) was used to fill the gaps of missing values for that same gene and group. For the second method, missing values were replaced with Cq value 40, which is the typical maximum Cq value and qPCR cycle amount. These are both common methods for replacing missing values, but more commonly used when a lower amount, perhaps only a single value, is missing and the method is not expected to introduce bias (McCall et al., 2014). The original data also include data points equal to 40, which might be due to the original researcher added those. Hence, there were two mean imputation datasets generated; one where Cq values of 40 were kept and one where these values were removed before imputation.
Microarray data

Microarray data was obtained via filtered searches on NCBI GEO DataSets (figure 3). “Colon cancer” was used as main search term, and results were filtered out to only show datasets where microarrays were used for expression profiling on samples from human tissue.

Out of the 41 results filtered out there were fifteen series including datasets that could be of interest. Further examination of the datasets showed that seven of these, despite filtering, had properties not suitable for further analysis; such as samples obtained from mouse xenografts, and another type of array being used than Affymetrix Human Genome U133 Plus 2.0 Array ([HG-U133_Plus_2] Affymetrix Human Genome U133 Plus 2.0 Array, 2018). Out of the remaining datasets one was randomly chosen for initializing the microarray data processing. The GSE44861 dataset was chosen, which included samples from both healthy human tissue as well as colon cancer tumor tissue (Affymetrix expression data from colon cancer patient tissues, 2018) (Ryan et al., 2014).

Despite using the “getGEOSuppFiles” function the raw data could not be downloaded via R, so it was manually downloaded from the GEO site. In the first implementation of the method processed data was used. However, the “getGEO” function (appendix A) stopped working after a couple of runs. A patch was loaded to counter the problem, however, it did not fix the problem and therefore raw data was used for further analysis.

The “biomaRt” package (Durnick et al., 2018a) was used to obtain the probe annotations matching the gene symbols in the qPCRset (appendix A). The annotations from the HG-U133_Plus_2 array were exported to to an .xlsx file and sorted into their matching genes (figure 9).

Since there are several probes on the array that can match a single gene (though different splice variants) a way to match values to the probe annotations that were in turn sorted under its associated genes was programmed (appendix A). There are no given guidelines to choosing which of the probes best matches the gene when information on transcript variants is unavailable. It is however recommended to not use the average of the expression values, and therefore the median expression value was used instead.
Figure 4. Snippet of gene median values. At the top microarray probe annotations are sorted under their matching genes. At the bottom the corresponding median microarray expression values are substituted for both control and target samples.

The median values for each probe set were obtained by sorting out the microarray control and target samples into different data sets, and calculating the average expression value for each probe in each respective data set. The average probe values were grouped to their matching genes, and the median function was applied for each gene (figure 4) (appendix A).

**Normalization and Combining Data**

From the qPCR data there are four datasets generated; one with the unaltered data (referred to as the raw dataset), one where NA’s were replaced with Cq value 40, one with mean imputed data still including the Cq value 40s from the original data, and a final one where all 40 values were removed and mean imputation was performed. These datasets were normalized using the well-established 2^ΔΔCt method (Rao et al., 2013). The 2^ΔΔCt values for each dataset were put into separate sheets and saved as tab delimited .txt files.

Two more datasets were created by adding the previously obtained median microarray expression values (figure 4) to the NAs in the raw dataset; in the first set microarray values replaced the NAs in the raw data and in the second dataset the microarray data replaced the missing ΔΔCt values from the original normalization. A full 2^ΔΔCt normalization was performed on these two datasets as well.

Additionally to exporting the normalized data two pheno data .txt files were created, simply determining which genes and samples are controls and which are targets. The first chunk of code was used to set up the pheno data used for all the datasets (appendix A). The second chunk (from #Raw expression data and down) is the part of the code which constructs the eSets, and it is repeated for
each of the datasets (table 2). Shown in the figure (appendix A) is how the eSet for the normalized raw data, referred to as simply “Expression”, was generated.

Table 2. Description of the normalized datasets. Description simply refers to which type of treatment has been performed on the named dataset, “raw data” simply refers to the eSet where no NA handling method has been used.

<table>
<thead>
<tr>
<th>Normalized Dataset Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Expression</td>
<td>Raw data</td>
</tr>
<tr>
<td>Expression NA=Ct40</td>
<td>Raw data NA’s replaced with Cq value 40</td>
</tr>
<tr>
<td>Expression w MiArr values</td>
<td>Raw data NA’s replaced with microarray data</td>
</tr>
<tr>
<td>Expression w MiArr DDCt values</td>
<td>Raw data ΔΔCt NA’s replaced with microarray data</td>
</tr>
<tr>
<td>Expression mean imputation</td>
<td>Mean imputed data with Cq value 40 removed</td>
</tr>
<tr>
<td>Expression mean imputation NA=Ct40</td>
<td>Mean imputed raw data</td>
</tr>
</tbody>
</table>

The standard deviation and mean values for the expression in each eSet (table 3) were generated and exported to .xlsx files, sorted into a common file and imported back to R. For the four DLG genes expression data was obtained for the control samples from each of the eSets. Appendix A shows the code used for performing statistical testing on DLG2 expression data. This process was repeated on each DLG gene, excluding the “Expression” set since the presence of NAs in the data prevents such calculation. For each of the genes standard deviation data was calculated, and the statistical tests ANOVA (table 4) and Tukey’s range post-hoc were performed (table 5). A subset of genes were chosen for closer analysis since presenting full analysis results of all genes from the qPCR data would be unnecessarily detailed, as well as take up too much space in the report, for giving an overview of how the different data processing methods compare. The DLG genes were chosen for this further analysis due to the involvement in cancer research as well as a close functional relation to each other (Subbaiah et al., 2012). The four genes do show some difference in standard deviation between each other after processing, as well as differing somewhat in the amount of NA’s present for each gene in the raw data (figure 2).

Table 3. Mean and Standard Deviation of expression sets. Differences in mean and SD values measured between eSet where no NA handling has been performed and each other respective dataset.

<table>
<thead>
<tr>
<th></th>
<th>Mean</th>
<th>Mean diff.</th>
<th>Standard Deviation</th>
<th>SD diff.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Expression</td>
<td>1.24</td>
<td>0</td>
<td>1.08</td>
<td>0</td>
</tr>
<tr>
<td>Expression NA=Ct40</td>
<td>1.28</td>
<td>0.04</td>
<td>1.73</td>
<td>0.65</td>
</tr>
<tr>
<td>Expression w/ MiArr values</td>
<td>1.18</td>
<td>-0.07</td>
<td>1.37</td>
<td>0.29</td>
</tr>
<tr>
<td>Expression w/ MiArr DDCt values</td>
<td>0.75</td>
<td>-0.49</td>
<td>1.27</td>
<td>0.18</td>
</tr>
<tr>
<td>Expression mean imputation</td>
<td>1.22</td>
<td>-0.02</td>
<td>1.33</td>
<td>0.24</td>
</tr>
<tr>
<td>Expression mean imputation NA=Ct40</td>
<td>1.41</td>
<td>0.17</td>
<td>1.62</td>
<td>0.54</td>
</tr>
</tbody>
</table>

Standard deviation and mean calculation were performed on the entire expression sets to give an overview of potentially large differences in the data shown before and after processing and NA handling (table 3). Naturally there are difference in expression between the control and target samples in several of the genes, therefore studying the SD and mean of a single dataset will not provide any
applicable information. However, since the same calculation is performed on all six datasets, the differences in mean and SD can be used to give an overview of how large effect the different processing methods have on the raw data.

The ANOVA (as well as the post-hoc testing) showed no significant differences in any of the DLG genes except for DLG4 (table 5), the gene with the most NAs of the four. For DLG genes 1 and 2 the adjusted p-values equals to, or are very close to, 1. The small difference between the two (DLG1 and DLG2) are likely due to the low amount of NAs in the target samples. DLG1 has the same amount of NAs as DLG3, and for the same samples, but there is a slight difference in adjusted p-values between the two. This might be due to differences in present expression values which has been used for processing.

Table 4. ANOVA statistical analysis of the four DLG genes. DF: degrees of freedom. Sum Sq: Sum of squares. Mean Sq: mean square. Pr(>F): p-value.

<table>
<thead>
<tr>
<th></th>
<th>Df</th>
<th>Sum Sq</th>
<th>Mean Sq</th>
<th>F value</th>
<th>Pr(&gt;F)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>DLG1</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Group</td>
<td>4</td>
<td>0.19</td>
<td>0.04822</td>
<td>0.32</td>
<td>0.865</td>
</tr>
<tr>
<td><strong>DLG2</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Group</td>
<td>5</td>
<td>0.17</td>
<td>0.0341</td>
<td>0.108</td>
<td>0.99</td>
</tr>
<tr>
<td><strong>DLG3</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Group</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td><strong>DLG4</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Group</td>
<td>4</td>
<td>137.3</td>
<td>34.32</td>
<td>57.22</td>
<td>&lt;2e-16 ***</td>
</tr>
</tbody>
</table>

Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

There are significant differences between all datasets except when comparing the two sets containing microarray data as well as comparing the mean imputation datasets (table 4) (figure 15). This indicates that when handling larger amounts of NAs there are differences between the different types of NA handling methods, but there is not a significant difference between the different ways the methods are used. Meaning there is not a large difference in the mean imputation of raw data with or without Cq value 40, as well as no difference when replacing NAs with microarray data on a raw data level or ΔΔCt data level.
Table 5. Tukey multiple comparison post-hoc results of DLG4 gene. Diff: difference between means of the two groups. Lwr: lower end point of CI. Upr: upper end point of CI. P adj: p-value after adjustment for multiple comparisons.

### Tukey multiple comparisons of means, 95% family-wise confidence level

Fit: `aov(formula = Expression ~ Group, data = DLG4_data)`

<table>
<thead>
<tr>
<th>Group</th>
<th>Diff</th>
<th>Lwr</th>
<th>Upr</th>
<th>P adj</th>
</tr>
</thead>
<tbody>
<tr>
<td>Expression mean imputation w NA=Ct40 – Expression mean imputation</td>
<td>-0.15955205</td>
<td>-0.5352097</td>
<td>0.2161056</td>
<td>0.7711449</td>
</tr>
<tr>
<td>Expression w DeltaDeltaCt Microarray Data – Expression mean imputation</td>
<td>0.97128188</td>
<td>0.5956243</td>
<td>1.3469395</td>
<td>0.0000000</td>
</tr>
<tr>
<td>Expression w Microarray Data – Expression mean imputation</td>
<td>0.91662347</td>
<td>0.5409659</td>
<td>1.2922811</td>
<td>0.0000000</td>
</tr>
<tr>
<td>Expression w NA=Ct40 – Expression mean imputation</td>
<td>1.59558870</td>
<td>1.2199311</td>
<td>1.9712463</td>
<td>0.0000000</td>
</tr>
<tr>
<td>Expression w DeltaDeltaCt Microarray Data – Expression mean imputation w NA=Ct40</td>
<td>1.13083393</td>
<td>0.7551763</td>
<td>1.5064915</td>
<td>0.0000000</td>
</tr>
<tr>
<td>Expression w Microarray Data – Expression mean imputation w NA=Ct40</td>
<td>1.07617551</td>
<td>0.7005179</td>
<td>1.4518331</td>
<td>0.0000000</td>
</tr>
<tr>
<td>Expression w NA=Ct40 – Expression mean imputation w NA=Ct40</td>
<td>1.75514074</td>
<td>1.3794831</td>
<td>2.1307984</td>
<td>0.0000000</td>
</tr>
<tr>
<td>Expression w Microarray Data – Expression w DeltaDeltaCt Microarray Data</td>
<td>-0.05465842</td>
<td>-0.4303160</td>
<td>0.3209992</td>
<td>0.9946272</td>
</tr>
<tr>
<td>Expression w NA=Ct40 – Expression w DeltaDeltaCt Microarray Data</td>
<td>0.62430682</td>
<td>0.2486492</td>
<td>0.9999644</td>
<td>0.0000717</td>
</tr>
<tr>
<td>Expression w NA=Ct40 – Expression w Microarray Data</td>
<td>0.67896523</td>
<td>0.3033076</td>
<td>1.0546228</td>
<td>0.0000115</td>
</tr>
</tbody>
</table>

There are differences, even though not significant, in average expression of the DLG1 (figure 5) and DLG2 (figure 2) in the data where NAs have been replaced with microarray data on the $\Delta\Delta$Ct level of normalization compared to the other datasets. This way of NA handling has had an effect on the control samples of genes where NAs were not present. Comparing this to the complete lack of any differences between the datasets seen in the DLG3 gene (figure 7), where the NA handling seemingly has not affected the data at all. The DLG4 gene expression set included several NAs in the control samples (figure 2) which makes mean imputation a poor choice for NA handling. This is shown in the boxplot (figure 8) with the very low expression values seen where mean imputation has been used. An average expression close to that seen throughout the data sets in the DLG1 and DLG3 genes is shown in both datasets including microarray data for DLG4. The data where NAs have been replaced with Cq value 40 show much higher expression than the other datasets, which is contradictory to a higher Cq value resulting in lower end analysis expression.
Figure 5. Boxplot of average normalized control expression for DLG1 gene. Mean expression shown on y-axis, datasets listed on x-axis. MI: Mean imputation. MI NA=Ct40: Mean imputation with NA=Ct40. MiArr DDCt: Expression with microarray ΔΔCt values. MiArr: Expression with microarray data. NA=Ct40: Expression with NA’s replaced with Cq value 40.

Figure 6. Boxplot of average normalized control expression for DLG2 gene. Mean expression shown on y-axis, datasets listed on x-axis. Raw: raw expression. MI: Mean imputation. MI NA=Ct40: Mean imputation with NA=Ct40. MiArr DDCt: Expression with microarray ΔΔCt values. MiArr: Expression with microarray data. NA=Ct40: Expression with NA’s replaced with Cq value 40.
Figure 7. Boxplot of average normalized control expression for DLG3 gene. Mean expression shown on y-axis, datasets listed on x-axis. MI: Mean imputation. MI NA=Ct40: Mean imputation with NA=Ct40. MiArr ΔΔCt: Expression with microarray ΔΔCt values. MiArr: Expression with microarray data. NA=Ct40: Expression with NA’s replaced with Cq value 40.

Figure 8. Boxplot of average normalized expression for DLG4 gene. Mean expression shown on y-axis, datasets listed on x-axis. MI: Mean imputation. MI NA=Ct40: Mean imputation with NA=Ct40. MiArr ΔΔCt: Expression with microarray ΔΔCt values. MiArr: Expression with microarray data. NA=Ct40: Expression with NA’s replaced with Cq value 40.
Discussion

qPCR data quality control

The provided qPCR data originally included a lot of not applicable values (NAs) which makes expression analysis of any of the genes where NAs are present not doable. This is because the HTqPCR package makes use of average cycle threshold (Cq) values of both groups of samples (control/target) for a gene for further processing and analysis (Dvinge and Bertone, 2018). The $2^{-\Delta\Delta C_{t}}$ normalization method also requires that there are no empty data points for the genes which are to be analyzed (Livak and Schmittgen, 2001).

To reduce the amount of computational processing and repetitive code needed for complete testing of all genes in the dataset a smaller set of genes were chosen for further analysis. The DLG genes were chosen due to their close relation, their role in colorectal cancer, and the varied amount of NAs present between the genes (figure 2) (Subbaiah et al., 2012). In future research a function should be created to make processing semi-automated allowing for more genes to be analyzed in a more convenient way. A larger set of expression values from more genes would enable for a more conclusive analysis.

Microarray data collection

The microarray data search was based on a flow chart of search terms and filters from previous research (Shangkuan et al., 2017). Despite specific filtering of the search results some of the downloaded datasets were removed from further analysis after manual review. Out of the remaining seven (7) datasets the “GSE44861” dataset was chosen for qPCR data completion because samples were obtained from similar conditions of the qPCR samples, and they were divided into two distinct groups; adjacent non-tumor colon tissue from patient with colon cancer (control) and tumor colon tissue from patient with colon cancer (target). In the initially planned research method it was described that multiple microarray datasets were to be used for data processing, however only one was used due to time constraints, and to be able to test the planned method. However, when improving the method it should be taken into consideration to use multiple datasets to improve on the microarray data.

Handling NAs and combining data

NAs were processed by different methods; replacing NAs with Cq value 40, mean imputation, and replaced with microarray expression values for the respective genes (McCall et al., 2014). Standard deviation (SD) and mean of the expression values of the entire generated datasets were obtained to compare how much the data changed with the different processing methods, so no consideration was taken to sort out the different groups of samples (target/control) and potential differences for the two.

The mean value differs very little between the pre developed methods for the raw expression dataset (table 3). The mean expression value for the two datasets where NAs are replaced with microarray data differs a lot between each other. The first set where microarray data replaces Cq value data NAs show little difference to the raw data expression set. Compared to the second microarray dataset where the mean value is lower than any of the other datasets. The least difference in SD compared to the raw expression dataset is observed in the second dataset with microarray data replacement, whereas the greatest difference to the raw data is present in the dataset where NAs are replaced with Cq value 40.

The DLG1 and DLG3 genes contained one (1) NA in each of their respective target sample group, whereas there were no NAs in the corresponding groups in the DLG2 and DLG4 genes. There were no NAs in the control sample group of any of the genes except for the DLG4 gene where 16 NAs were...
The statistical testing showed significant differences between the datasets only in the DLG4 gene, whereas all the other DLG genes showed close to no difference between the datasets. The differences seen in the DLG4 gene (figure 8) compared to the very similar data observed in the DLG1 and DLG3 genes likely reflect the higher impact missing values in the control samples group has on normalization compared to missing values in the target samples. The average ΔCt step of 2^{-ΔΔCt} takes average values of only control samples, which may explain why as to the genes where target sample values are missing are not as affected compared to when NAs are present in the control samples. Differences and expression patterns shown throughout the datasets in all four DLG genes for the target samples are very similar to the control samples.

A slight, insignificantly lower average expression is observed in the DLG2 gene in the dataset where microarray data has been added at the ΔΔCt level (figure 6). The same difference is seen in both control and target samples. Even if it is insignificant, this difference in expression is strange since no NAs are present in any of the sample groups of the gene, so there should be no difference at all between the datasets since none of the NA handling methods have been used.

Results in relation to the aim

Three approaches to integrating microarray and qCPR data were discussed in the method description; modifying pre-developed integration methods to fit the data, replacing NAs with microarray data, and replacing qPCR control data with microarray data. Here, initialized testing of the second approach has been performed. While no concluding method has been developed, an investigatory foundation for testing the compatibility between the two data types has been laid by replacing NAs in a small subset of genes and analyzing the control samples. A lot of planned work was delayed and not fitted into the time frame of the research because of existing packages and functions not working the way intended, or not working at all.

Methods and conclusions in scientific context

In addition to previously developed methods for NA handling (mean imputation and replacing them with Cq value 40) a novel method has been developed and suggested where missing qPCR values are replaced with corresponding microarray expression values. While well implemented in previous studies the both mean imputation and Cq value 40 methods are very bias prone and only recommended if used to replace a single value in a sample set (McCall et al., 2014) (Chowdhry et al., 2016).

There are similarities in the results generated from all the NA-handling methods in the genes where missing values are observed only in the target samples. These missing values are however few, and the differences between the methods are likely insignificant due to the low impact a single missing value has on a large dataset (Dong and Peng, 2013). There is a more significant change in obtained values in the DLG4 gene (figure 3). Missing values in this gene differs from the other observed NAs in two aspects; they are observed in a higher amount, and they are all found in the control sample data. When returning to this research a larger set of genes should be analyzed to assess the impact different sets of missing values has on the data through the different methods. These sets include those seen in this research; no NAs, a few NAs only in target samples, a larger amount of NAs in control samples. Sets should also include varied amounts of NAs seen throughout both sample groups (control/target).

The more recommended method, compared to mean imputation and replacing NAs with Cq 40, for dealing with non-detects is to use the functions included in the “nondetects” package (McCall and Sherina, 2018). The obtained data did not fit the package function in any of the tried configurations,
and not enough information was provided via reference manual to easily figure out the correct setup for the package. If the demonstrated methods are revisited effort should be put towards making nondetects functional with the present data. This is to be able to compare the proposed method (replacing missing values with microarray data) with a more robust NA handling method than used in this research.

In short, while no significant conclusion is drawn from the results, the research shows a novel approach to integrating microarray and qPCR data by initially replacing NAs in qPCR data with microarray data. The potential data compatibility is demonstrated by the minor SD and mean differences shown when comparing the suggested method to the previously established methods.

Novelty

While the compatibility of microarray and qPCR data has been reviewed in previous research, microarray data has not been used to fill missing values in qPCR data (Allanach et al., 2008).

Previous research has focused on comparing microarray and qPCR technologies by outcome similarities in experiments testing the different methods on a similar set of samples. This includes (but is not limited to) looking at expression levels of a single gene and compare the results with immunohistochemistry (IHC) of the same gene and tissue, identifying histopathology diagnostic genes in a larger set of genes, as well as identifying microRNA (miRNA) profiles in histology studies (Allanach et al., 2008) (Bergqvist et al., 2007) (Camarillo et al., 2011). These approaches differs from this research, where a novel method of applying microarray data to fill gaps in qPCR data is tried.

Ethical aspects and impact on society

Both types of data (microarray/qPCR) were obtained from external sources, so no consideration had to be taken towards ethical implications of obtaining the original clinical samples. None of the provided data has been used with the intention to in any way identify the patients from which the samples were obtained from.

If publically available microarray data could be used to improve on the quality of poor qPCR data it could mean that research where repeating experimental procedures is unfeasible or too resource consuming to do could still lead to more conclusive results.

Future directions

This research functions as a pilot study and initial step towards the possible integration of microarray and qPCR data. When returning to the project the code needs to be revised and the original data looked over and fitted to, first of all, the nondetects package (McCall et al., 2014). A comparison of the proposed method of handling NAs towards a method preferred over the mean imputation and Cq value 40 used would add towards a more robust analysis. Continuing from this a method may be built to randomly replace present qPCR data with microarray expression data to see if any differences are shown after data replacement compared to before. If there is good compatibility between the two data types, with minor or no differences seen when replacing qPCR data with microarray data the two other approaches (not tested in this research) may be tested. First overall qPCR control sample data may be replaced with microarray data, before an integration approach based on previous integration models is tested.
Some of the packages and functions suggested in the planned method has not worked at all or not functioned as simply as described, which led to more time had to be put towards creating original code than initially planned. This in turn led to a not as in depth analysis was performed as would have been preferred due to time constraints. In future research a potentially useful premade function or package should be researched not only by looking at the original article where the package or function in question is introduced and demonstrated, but looking at citations, as well as amount of citations, to assess to which extent it can be used with minimal manipulation. A pilot study can be conducted to test the initialized use of the different methods, to try to predict if a lot of resources have to be put towards adapting the available datasets to fit the potential methods.

This research has focused on improving on quality of a qPCR dataset obtained at a post-wet lab state of an expression analysis. No information has been provided on the possible reasons for why the different NAs have been present, just that there is missing values due to either low expression or experimental errors. If it could be determined if an NA is the result of low expression of the given gene in a sample it could be motivation to use a simple method of NA handling such as setting expression to 40. If an experiment could be designed to test the Cq curves in a dataset, where some of the samples have been intentionally put through experimental errors and samples with known low expression of a certain gene are included, to a regression model to see if there is any differences which characterizes one or the other.

As with any research which includes multiple parties from different disciplines of the biomedical research field all parties should be included in the initial planning to minimize the risk that any steps during the research are overlooked. As demonstrated in this research a lot of time could have been saved if information on the gene transcripts were saved during the qPCR analysis, making the need for sorting out median expression values of the probe sets of interest redundant. In future research, if a bioinformatic analysis is needed any time during the research, a bioinformatician should be consulted during the planning stage, so to inform on and highlight on information that needs to be saved throughout the research up to the bioinformatic analysis.

Acknowledgements

I, the author, would like to thank Angelica Lindlöf for great supervision and feedback, and examiner Björn Olsson for his feedback as well. I would especially like to thank Hendrik de Weerd for helping with the code, specifically the part sorting probe ID’s into their corresponding genes. I thank Simon Keane for the qPCR data and for once again letting me be part of his research. Additionally Benjamin Ulfenborg is thanked for his quick response to my questions regarding some of his R scripts. Finally my most heartfelt thanks go to Anna for once again supporting her boyfriend through five months of complete asocial work.
References


Allanach, K., Department of Medicine, D. o. N. I., Alberta Transplant Applied Genomics Centre, Mengel, M., Department of Medicine, D. o. N. I., Alberta Transplant Applied Genomics Centre, Einecke, G., Department of Medicine, D. o. N. I., Alberta Transplant Applied Genomics Centre, Sis, B., Department of Medicine, D. o. N. I., Alberta Transplant Applied Genomics Centre, Department of Laboratory Medicine and Pathology, U. o. A., Edmonton, Canada, Hidalgo, L. G., Department of Medicine, D. o. N. I., Alberta Transplant Applied Genomics Centre, Mueller, T., Department of Medicine, D. o. N. I., Alberta Transplant Applied Genomics Centre, Halloran, P. F. and Department of Medicine, D. o. N. I., Alberta Transplant Applied Genomics Centre (2008) 'Comparing Microarray Versus RT-PCR Assessment of Renal Allograft Biopsies: Similar Performance Despite Different Dynamic Ranges', American Journal of Transplantation, 8(5), pp. 1006-1015.


Dvinge, H. and Bertone, P. (2018) HTqPCR: High-throughput analysis and visualization of quantitative real-time PCR data in R *Bioinformatics*. Analysis of Ct values from high throughput quantitative real-time PCR (qPCR) assays across multiple conditions or replicates. The input data can be from spatially-defined formats such ABI TaqMan Low Density Arrays or OpenArray; LightCycler from Roche Applied Science; the CFX plates from Bio-Rad Laboratories; conventional 96- or 384-well plates; or microfluidic devices such as the Dynamic Arrays from Fluidigm Corporation. HTqPCR handles data loading, quality assessment, normalization, visualization and parametric or non-parametric testing for statistical significance in Ct values between features (e.g. genes, microRNAs). Available at: https://www.bioconductor.org/packages/release/bioc/manuals/HTqPCR/man/HTqPCR.pdf (Accessed: 2018.02.18).


Appendix A

```r
# setting up work directory  
setwd("~/R/DATA.DIR/Thesis/renamed_samples/")  
library("HTqPCR")  
path<="C:/Users/jonas/Documents/R/DATA.DIR/Thesis/renamed_samples/")  
# reading in raw data  
files<-read.delim("files.txt")  
# setting up qPCRset  
rawtest <- readCtData(files=files$Files, path=path,  
n.features=51,header=FALSE,n.data=1, na.value=NA)  
express<- rawtest@assayData$exprs  
View(express)  
rownames(express)<-rawtest@featureData$data$featureNames  
g<-featureNames(rawtest)[1:16]  
plotCtOverview(rawtest,genes=g,xlim=c(0,50),groups=files$Type,conf.int=TRUE,ylim=c(0,50))  
raw.cat<-rawtest  
plotCtCategory(raw.cat,by.feature=TRUE,cexRow=1.0,cexCol=1.0)
```

Figure 9. Code used to generate and visualize the qPCRset.

```r
library("nondetects")  
pData(rawtest)$sampleName<-colnames(exprs(rawtest))  
tst<-qpcrImpute(rawtest) 
~0 + nrep  
<environment: 0x00000000053f3cb8>  
Error in Name.s2[ind2] <- names(tst$sigma[i]) :  
  replacement has length zero
In addition: Warning message:  
glm.fit: algorithm did not converge
```

Figure 10. Code used for initializing the nondetects package.

```r
library("GEOquery")  
library(oligo)  
setwd("~/R/DATA.DIR/Thesis/Microarrays/")  
GSE44861data<-getGEO("GSE44861",  
destdir="C:/Users/jonas/Documents/R/DATA.DIR/Thesis/Microarrays/GSE44861/" )  
# Create eSet from pre-processed data  
eset<-GSE44861data$GSE44861_series_matrix.txt.gz  
# Create eSet from raw data  
untar("GSE44861/GSE44861_RAW.tar",exdir="CEL_dat")  
cel.dat<-read.celfiles(filenames=paste("CEL_dat/",list.files("CEL_dat"),sep=""))  
raw.eset<-rma(cel.dat)
```

Figure 11. Code used for creating eSet from microarray data.
# Setting up gene identifier
library("biomaRt")
ensembl<--useMart("ensembl")
ensembl<--useDataset("hsapiens_gene_ensembl",mart=ensembl)

# Specifying query
print functions:
genefile<-read.delim(file.path("C:/Users/jonas/Documents/R/DATA.DIR/Thesis/Microarrays/gene_annotations.xlsx",sheetName=genefile)
genefile<-genefile$Genename_annot<-getBM(attributes=c("affy_hg_u133_plus_2", "hgnc_symbol", "entrezgene", "chromosome_name", "start_position", "end_position", "band"), filters="hgnc_symbol", values=genes, mart=ensembl)

deleting annotation
library(xlsx)

Figure 12. Code for generating correct gene annotations in the microarray.

# List and filter annotations
genefiltered<--exprs[rownames(exprs) %in% geneannot$affy_hg_u133_plus_2, ]
# setting class vector and separating expression in control and target
cv<-read.delim("GSE44861/class_vector.txt")
target_exprs<-genefiltered[,c(cv$Target)]
control_exprs<-genefiltered[,c(cv$Control)]
control_exprs<-control_exprs[,ncol(control_exprs)]

# Averaging values
avrg.control_exprs<-as.matrix(rowMeans(control_exprs))
colnames(avrg.control_exprs)<-"Mean Value"

avrg.target_exprs<-as.matrix(rowMeans(target_exprs))
colnames(avrg.target_exprs)<-"Mean Value"

# Importing sorted annotations
sorted_annot<-read.delim("gene_annotations_sorted.txt")

# Listing annotation values
ctrl<-as.vector(avrg.control_exprs)

names(ctrl)<-rownames(avrg.control_exprs)
cols<-sapply(x = names(ctrl), FUN = function(x) (which(sorted_annot == x, arr.ind = T)[,2]))

colvector<-unlist(cols)
sorted.avrg.control_exprs<-lapply(X = 1:max(colvector), FUN = function(x){ctrl[colvector==x]})

trgt<-as.vector(avrg.target_exprs)
names(trgt)<-rownames(avrg.target_exprs)
cols<-sapply(x = names(trgt), FUN = function(x) (which(sorted_annot == x, arr.ind = T)[,2]))


colvector<-unlist(cols)
sorted.avrg.target_exprs<-lapply(X = 1:max(colvector), FUN = function(x){trgt[colvector==x]})

# Generate median values
control_median_values<sapply(X = sorted.avrg.control_exprs, FUN = median, na.rm = T)
target_median_values<sapply(X = sorted.avrg.target_exprs, FUN = median, na.rm = T)

# Export median values
write.xlsx (target.median.values,"C:/Users/jonas/Documents/R/DATA.DIR/Thesis/Microarrays/target_median.xlsx", SheetName=gene_annot, showNA=FALSE)
write.xlsx (control.median.values,"C:/Users/jonas/Documents/R/DATA.DIR/Thesis/Microarrays/control_median.xlsx", SheetName=gene_annot, showNA=FALSE)

Figure 13. Code for obtaining median expression values for each gene.
#construct esets
library("Biobase")
setwd("R/DATA.DIR/Thesis/Delta/")
#Setting up metadata
pDataFile <- "Pheno_Data.txt"
pData <- read.table(pDataFile, row.names=1, header=TRUE, sep="\t")
metadata <- data.frame(labelDescription="Tumor/Control status", row.names="Type")
phenoData <- new("AnnotatedDataFrame", data=pData, varMetadata=metadata)
gpData <- read.table("Gene_pData.txt", row.names=1, header=TRUE, sep="\t")
gmetdata <- data.frame(labelDescription="Target/Control gene", row.names="Type")
GenePhenoData <- new("AnnotatedDataFrame", data=gpData, varMetadata=gmetdata)
genes <- read.delim("C:/Users/jonas/Documents/R/DATA.DIR/Thesis/Control_Values/Gene_pData.txt")
genes <- genes$Gene
sum_colnames <- c("Mean", "Standard Deviation")
#Raw expression data
Expression <- "Expression.txt"
exprs_Expression <- as.matrix(read.table(Expression, header=TRUE, sep="\t", row.names=1, fill=TRUE))
x <- colnames(exprs_Expression)
i <- gsub("\", " ", x)
colnames(exprs_Expression) <- i
all(rownames(pData) == colnames(exprs_Expression))
nas <- which(is.na(exprs_Expression), arr.ind=TRUE)
na_row <- which(genes %in% rownames(nas))
exprs_Expression_rm_na <- exprs_Expression[-c(na_row),]
gpData_rm_na <- as.data.frame(gpData[-c(na_row),])
rownames(gpData_rm_na) <- rownames(exprs_Expression_rm_na)
colnames(gpData_rm_na) <- "Status"
GenePhenoData_rm_na <- new("AnnotatedDataFrame", data=gpData_rm_na, varMetadata=gmetdata)
eSet_Expression <- ExpressionSet(assayData=exprs_Expression_rm_na, phenoData=phenoData, featureData=GenePhenoData_rm_na)
eSet_Expression <- outlier2na(eSet_Expression)
eSet_Expression <- mean.imputation(eSet_Expression)
exprs_Expression_mean <- mean(eSet_Expression$assayData$exprs)
exprs_Expression_sd <- sd(eSet_Expression$assayData$exprs)
exprs_Expression_sum <- data.frame(exprs_Expression_mean, exprs_Expression_sd)
colnames(exprs_Expression_sum) <- sum_colnames
rownames(exprs_Expression_sum) <- "Expression exprs"

Figure 14. Code used to create eSets from the normalized data.
Figure 15. Code used to perform statistical analysis on the DLG2 gene.