Internal Report 2012–03 June 2012

Universiteit Leiden
Opleiding Informatica

Physiological ageing described
by genome-wide expression trends

Marius Gheorghe

MASTER'S THESIS

Leiden Institute of Advanced Computer Science (LIACS)
Leiden University
Niels Bohrweg 1
2333 CA Leiden
The Netherlands

Physiological ageing described by genome-wide expression trends

Master Thesis
LIACS - Leiden University

Author:
Marius Gheorghe

Coordinator:
Dr. Vered Raz
(LUMC)

Supervisors:
Dr. Michael Emmerich
Prof. Dr. Thomas Bäck
(LIACS)

Abstract

Biomedical research uses gene expression in order to understand biological processes and diseases. The level of gene expression is subject to change during the lifetime of an individual and this deregulation can reflect ageing-associated disorders and decline in physical ability. By studying the genome-wide expression trends of the human genes, underlying age-associated molecular processes may be identified and further understood. To perform an adequate analysis, different data mining and pattern recognition concepts are to be applied in the data processing. This paper describes a step-by-step methodology developed to perform longitudinal gene expression microarray dataset analysis in order to identify such age-associated expression trends and determine their biological functionality. The analysis was performed on healthy skeletal muscle tissue samples and validated on several other tissues obtained from non chronically ill individuals. Significant ageing-associated gene clusters were identified as a result of the study.

Acknowledgments

I would like to thank Dr. Vered Raz, project coordinator from the Leids Universitair Medisch Centrum, department of Human Genetics, for offering me the chance to participate in such an interesting research topic and for the help provided during the development of the project. Moreover, Dr. Raz was involved in gathering and generating the skeletal muscle dataset that was used in this study. Her extensive knowledge in the field of molecular biology clarified and enhanced the understanding of an unbiased approach in the research and in the methods used. Furthermore, it facilitated the convergence of this computer science master project with the biology.
I would like to offer special thanks to Dr. Jelle J. Goeman from the Leids Universitair Medisch Centrum (LUMC), department of Medical Statistics who actively participated in the follow-up meetings of the project, clarifying the emerging issues. The undisputed experience of Dr. Goeman provided an insight of the statistical approach in a research project based on biological data and the level of severity of the applied methods.
Also, I would like to thank Andrea Venema from LUMC, department of Bioinformatics, for gathering and preprocessing the validation datasets and Dr. Yahya S. Anvar, department of Medical Statistics for the constructive discussions.
I wish to thank Dr. Michael T. M. Emmerich and Prof. Dr. Thomas Bäck, my project supervisors, for their contribution and the various approaches suggested to overcome the emerging issues, proof of their incontestable knowledge in the field of computer science.
Last but not least, I want to express my gratitude towards my parents for their absolute support throughout the master programme.
Thank you all once again for making this study possible.

Contents

List of Figures

List of Tables

1. Introduction

The expression level of a gene is controlled by its surrounding DNA sequences. Transcription factors (e.g., proteins) bind to these sequences causing a gene to switch on or off. The availability and the activity of the transcription factors thus regulates the expression level of a gene or a group of genes. Several human diseases result from the absence or malfunction of certain transcription factors which cause the disruption of gene expression. The underlying genetic mechanisms which lead to this natural variation in gene expression are a key factor in understanding how expression levels are regulated and how genes interact with each other. Moreover, it can provide a good insight of the trigger mechanisms for various diseases. In order to study this phenomenon, a quantification of the gene expression levels is necessary. This is achieved by transcribing the gene to yield a primary transcript which is later processed to remove the introns[1] and generate a mature transcript or messenger RNA (mRNA) that only contains exons. The mRNA is then used to generate a microarray[1] reflecting the expression levels of the genes. The dataset formed from the microarray represents the basis of this study. Identifying and analyzing age-associated gene expression trends requires several data processing steps, necessary for the robustness and consistency of the study. This paper presents each of the steps the methodology makes use of and their results evaluated.

1.1. Related work

Research related to the significance of the expression level changes in muscle tissue associated several genes with late onset muscular diseases or dystrophies[2][3] and triggering molecular mechanisms were identified[4][5]. The research is exhaustive and the yield of genes that may be associated to dystrophies or diseases is constant. Researchers regularly identify specific gene mutations that are responsible for muscle weakness and degeneration in their quest to fully

[1]The majority of human genes are divided into introns (i.e., intragenic region) and exons. Only the latter carries out information about the protein synthesis.

understand the primary causes of a disease. Previously, studies were performed on the expression trend of a gene or a group of genes, but not genome-wide, in order to identify, group and classify ageing-associated genes.

1.2. Hypothesis and approach

Muscle strength denotes a healthy ageing and it is suggested to predict disability and mortality in elderly individuals. The analysis of genome-wide expression trends may describe physiological ageing and thus help understanding its underlying molecular processes. Moreover, it may result in identifying genes or groups of genes responsible for certain muscular dystrophies and provide the basis for further research. To perform an unbiased and robust analysis, the microarray datasets are subject to applicability of various data mining and pattern recognition concepts. In this approach, the resulting microarray dataset was smoothened to reduce variation and filtered in order to discard the genes that do not present an important age-associated effect, prior to the identification of the major trends in the gene expression levels. The resulting gene clusters provide the necessary material for an analysis of their biological functionality. Each of the aforementioned steps is detailed and the choice of methods justified.

1.3. Outline

The generation of the data and the preliminary processing steps are described in the next chapter. Chapter 3 contains a detailed description of some of the best suited data smoothing methods for human gene expression analysis and provide a comparison of their applicability on the skeletal muscle tissue. Moreover, the data filtering step is also detailed. In the following chapter the different clustering methods that were applied on the data and their limitations are presented. Results of the identified major trends in the gene expression levels are depicted for various tissues and an interpretation provided. Chapter 5 contains the steps followed to determine the biological functionality of the resulting gene clusters and provide a visual example of their biological relevance. A conclusion will mark the end of the paper.

2. Preparing the data

In order to provide an unbiased and robust study of the genome-wide expression trends, several processing steps are required. First, the gene expression microarray dataset has to be generated from tissue samples. The microarray dataset is then subject to a series of data processing protocols with the aim of reducing the impact of the machines used in the process, identifying outliers and test its age-associated significance. These steps are detailed in the next section.

2.1. Data generation

The microarray datasets are mostly generated from biopsies. Samples of tissue are obtained from patients and gathered to later generate a digital database. Different institutions or submitters make this data available for gene expression studies. The microarray generation protocols may differ but the process remains mainly the same. The RNA is first isolated from different tissues and then labeled and hybridized[2] to the arrays which allows expression to be measured and compared between the appropriate sample pairs. The arrays are scanned and gray scale images are generated for each pair of samples to be compared. An analysis of the images is latter required to measure the fluorescence intensities of the arrayed spots. For a more detailed description of the process one is referred to [6].

2.2. Data preprocessing

The RNA can be spotted on more than one microarray which can be obtained from different batches. The dataset obtained from quadriceps (i.e., skeletal muscle) samples represents an example of this case. In order to properly merge

[2]Various methods are used to radioactively or chemically label the DNA. This process allows to visualize where the DNA hybrids are forming onto the carrier and to be compared with the reference DNA.

Tissue	Samples	Age range	GT p-value	Probes
quadriceps	29	35-89	0.015	48803
brain frontal cortex	30	26-106	0.004	12558
kidney cortex	72	27-92	0.002	22283
kidney medulla	61	29-92	0.205	22283

Table 2.1.: Details of the available datasets generated from various tissues.

the microarrays, VSN normalization[3] and batch effect correction (i.e., non-biological experimental variation) as well as covariate (e.g., gender) correction were applied on the microarrays. The reader interested in more details about these data preprocessing steps is referred to [7] as they do not represent the purpose of this paper.

Principal component analysis[4] or PCA for short, was performed on the datasets in order to identify and eliminate any outliers. A global test[8] (GT for short) was performed on the datasets in order to determine their level of significance via a p-value[5] in an age-associated context. Table 2.1 summarizes the details of the datasets used in this study. Note that only the datasets with a GT p-value < 0.05 are to be considered statistically significant [9] and thus included in further analysis. The second column contains the population size for each of the datasets and the fifth contains the number of probes that were spotted on the gene expression microarray. Note that a gene is expressed twice in average, thus for the quadriceps dataset only $\approx 25,000$ individual genes are expressed from an initial number of 48803 probes. In order to bring the datasets to a "cleaner" representation and to be able to discard the probes that do not present a relevant age-associated effect (i.e., not significant) the data has to be smoothened and filtered. These are the next two steps of the general workflow of the study, as depicted in the flow diagram in Appendix A. These steps are detailed in the next chapter, as well as the concept of *significant probe*.

[3] The variance of the microarray datasets is due to the signal intensity. Variance Stabilization and Normalization aims at finding a transformation that will render this variation approximately constant.

[4] A mathematical procedure which converts a set with possibly correlated variables into a set with uncorrelated variables called principal components. The first component has the highest variance. It is often used to reduce the dataset dimensionality.

[5] A p-value represents a probability measure (ranging from 0 to 1) of the measurement being the result of random sampling.

3. Data smoothing and filtering

Smoothing of the data is essential to reduce the variation between the individuals, thus providing a cleaner dataset for the proceeding steps in the analysis. This is achieved by bringing the extremes in the dataset to the mean. Several data smoothing methods were applied on the dataset obtained from skeletal muscle tissue in order to identify the best fitted method for gene expression analysis. The dataset does not have an uniform distribution nor a normal distribution according to the Chi-square and the Kolgomorov-Smirnow tests[4], the significant difference being around 99%. This is due to the existing gaps between the age of the individuals. No age redundancies are present as the exact lifetime of the individuals until the sampling was computed. The smoothing was carried out per probe on the dataset ordered ascendingly by age.
In the following pages, the different techniques of data smoothing are described:

3.1. Splines

A spline function is a piecewise polynomial function which passes through a defined number of control points or, in other words, is fitting a smooth curve to a set of noisy observations. Spline interpolation is preferred over simple polynomial interpolation, due to the possibility to minimize the interpolation error, even when using low degree polynomials for the creation of the fitting curve. For instance, quadratic splines use second-order polynomials in order to produce a sufficiently smooth spline. There are also cubic, quartic or quintic splines. For this study, several types of splines were tested. The control points (or knots) of the curve can be either predefined, either randomly picked. The choice of the number and position of the knots will also be discussed.

[4] The Kolmogorov-Smirnov and Chi-square are two "goodness-of-fit" non parametric tests that apply to statistical distribution. They are used to determine if the data points in a dataset are uniformly distributed or not.

Cubic splines

A cubic spline is a spline function that uses third-order polynomials in order to compute a sufficiently smooth curve to fit the noisy data. Apart the degree of the polynomials used in the spline, there are also several different types of splines. Among these are `natural, periodic` or `fmm` splines. The `natural` splines are curves which are forced through the knots, but allow the slope to be free at the ends in order to minimize the oscillatory behavior of the curve. The `periodic` spline is a spline that can be applied periodically, if the predicted curve has the tendency or is *closed*. The `fmm` spline or the spline of Forsythe, Malcolm and Moler [12] fits an exact cubic through the four points at each end of the data as a method to determine the end conditions. Of course, these are only a few examples. The interested reader is referred to [14]. The syntax of the cubic spline function is as follows:

cubic_ spline(x,y,spar,cv)

with x representing the vector of the abscissa (e.g., the age values) and y the vector with the corresponding gene expression levels. The *method* parameter refers to the different types of splines that can be applied on the data. The choice is between `periodic, natural` and `fmm`.
The *spar* parameter is the smoothing parameter. The coefficient of the integral of the squared second derivative in the fitness criterion is a monotone function of the *spar* parameter. Normally, its value is ranged between 0 and 1 but higher values can also be applied as it will be shown in the following lines. For more details regarding the algorithm behind a cubic spline function, one is referred to [12][13]. The *cv* parameter is a logical parameter that determines which type of cross-validation[5] is to be used. If set to `TRUE` then `ordinary` cross-validation is used against `generalized` cross-validation. The former uses a "leave-one-out" strategy which is just K-fold cross validation taken to its logical extreme, with K equal to the number of data points. In other words, the function approximator is trained on all the data except for one point and a prediction is made for that point. This process is repeated as many times as data points there are in the set. The latter one is a weighted version of the ordinary cross-validation. It is better suited for duplicates in the x vector. For the interested reader, more details about cross-validation can be found in [15].

[5] Cross-validation is an estimation method of how accurate the used prediction model is. In other words, it checks how well a model generalizes to new data. It also known as rotation estimation.

Note that the spline function is built in such a way that the abscissa values should be distinct and strictly increasing, one of the reasons why the exact age (i.e., using years and months) of the individuals was computed. With the *cv* parameter set to `TRUE` if duplicates occur in the x vector, the spline function choses only one of the elements in order to avoid a division by zero in the intermediate computation of the predicted points. The behavior of the function when more than two duplicates are present is even more ambiguous as groups of duplicates are left out. In other words, no rule is defined regarding the choice of the element to be taken in consideration. For the rest of this study the *cv* parameter will be set to `TRUE` as no duplicates are present in the x vector.

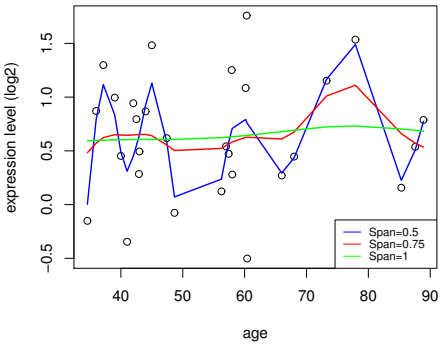


Figure 3.1.: Cubic spline applied on the gene HS.541791 from the dataset.

Fig. 3.1 shows the resulting fitted curve of a `natural` cubic spline on the gene HS.541791 from the dataset. This probe has not yet been identified by its functionality. One can clearly see that as the *spar* parameter is increased, the fitted curve has a more linear trend as its oscillation liberty is substantially reduced. The number of knots is inversely proportional to the *spar* parameter. This also contributes to the linear tendency of the fitted curve. It is possible to assign all the data points as knots for the spline function, but this will not produce any desired smoothing effect as it is the equivalent of connecting all the data points with a line. For the results presented in rest of the paper, the *spar* parameter was set to 1 as it displays the desired smoothing effect.

The `natural` spline is more adequate for this dataset, as the first data point of a probe does not represent the initial expression level of the gene (e.g., age 35), nor the last one represents the final expression level that a gene might have. In other words, the extremities of the fitted curve are not overlapping the extremities of the probe thus allowing the spline function to be more stable and behave more "natural". The `periodic` splines and the `fmm` splines are not suitable for this study as the former is more adequate for genes that express a periodicity in their expression level, through time and the latter tries to find the end conditions by fitting the curve through the end points of the data. As a conclusion, the *method* parameter was set to `natural`.

Bézier curves

A Bézier spline is a series of Bézier curves[6]. This parametric curves have multiple applications, computer graphics being one of the well-known and wide-spread domains. In its most common use it is a simple quadratic or cubic equation that performs a smoothing of the data, by "drawing" a curve through a specified number of control points. The shape of the curve and its liberty depends of course on the degree of the polynomials used in the interpolation. The above presented smoothing techniques are either a generalization, either a particular case of Bézier curves.
In this study, a cubic equation was used for the smoothing. The syntax of the function has the following form:

bezier(x,y,nbPoints)

with x representing the vector containing the abscissa values (e.g., age) and y a vector containing the corresponding gene expression level for each element of x. The *nbPoints* parameter sets the number of control points of the described curve. If *nbPoints* is set to two, then the resulting curve will be a line segment between the first and the last data point. As the number of control points is increased, the linear tendency of the fitting curve disappears. One of the draw-backs of a Bézier curve, in this context, is that the curve will always pass through the first and last control point (i.e., the first and last pair of (x,y) values), but almost never through the other control points, regardless the setting of the *nbPoints*. This has a significant impact in the resulting smoothened curve, as it is forced to pass through the extremities of the probe. In this study it was used as a reference to the other spline functions. For more details regarding the Bézier curves, one

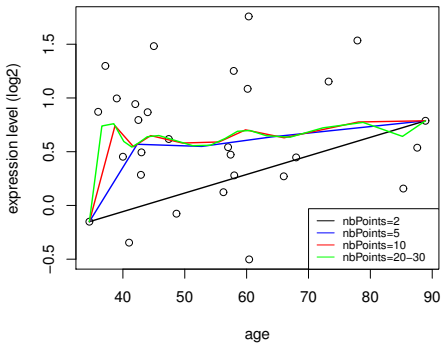


Figure 3.2.: Bézier curves applied on the gene HS.541791.

[6] A Bézier curve is a parametric curve that employs at least three points to define the curve. It is formed by two anchor points (first and last point of the curve) and control points which can alter the shape of the curve.

is referred to [16].
Fig. 3.2 shows the behavior of the Bézier curves with different number of control points. One can easily see how the fitting curves are always attached to the first and the last data point. As discussed in Section 3.1.1 this behavior is not suited for this study, but, as mentioned, it serves well as a reference in the search of the best spline configuration to be applied on human gene expression. Increasing the value of the *nbPoints* parameter results in a more smoothened curve that fits the noisy data. An overall observation showed that if the number of control points is set to any value bigger than $\approx 2/3$ of the total number of data points of a probe, the fitting curve does not present significant changes. In this particular case, the choice of 20 or more control points (out of 29 possible) did not yield any substantial modifications in the shape of the curve.

B-splines

Besides the natural cubic splines and the Bézier curves another type of splines was considered. The B-splines or *basis splines* are a type of splines for which a function basis is created. The spline basis controls the behavior of the curve, thus allowing the user to easily and consistently influence the fitting curve. Each control point of the spline is provided with a function in the spline basis, dictating the behavior of the curve when reaching that particular data point. Moreover, the degree of the polynomials used in the spline can also be defined by changing the functions in the spline basis.
In the following lines different spline basis will be discussed and compared in order to find the best configuration to be applied on the dataset. The syntax of the B-spline function is as follows:

bspline(x,y,degree,knots)

with x representing the abscissa values (e.g., age) and y contains the corresponding gene expression level for each element of x. The *degree* parameter sets the degree of the polynomials used in the spline function and the *knots* parameter defines the control points for the fitting curve.

Fig. 3.3 shows the behavior of the B-spline with different settings of the *degree* parameter. For this figure, three control points were defined, equidistant from the extremities of the probe and among each other. The fitting curve of the quartic (i.e., degree four polynomials) and cubic B-spline show no real tendency to smoothen the data due to the high-order polynomials. The distance between the predicted points of the fitting curve becomes too large within the neighborhood of the control points, thus reflecting a *zig-zag* behavior. Furthermore, a higher degree polynomial always has the tendency of "over-smoothing" the data. This phenomenon depends on the data distribution and the choice of the control points, but as stated the available dataset does not present a normal distribution of the data. In the example provided in Fig. 3.3 this effect is present in the very first part of the curve (e.g., left side of the plot), until reaching the first control point. This tendency of *closing* the curve is not suited for this study, as discussed in Section 3.1.1. The quadratic B-spline shows a better behavior than the quartic or cubic spline as the "zig-zag" effect is reduced, but it is still present. For further experiments, the quadratic and the linear B-splines were chosen from this group, as they reflect a behavior that is closer to the desired effect. An impact on the number of knots was studied in order to determine the right configuration of the *knots* parameter.

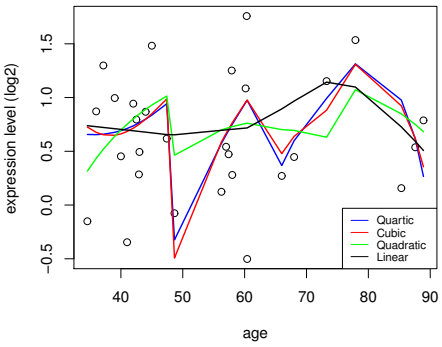


Figure 3.3.: B-spline function with different polynomial degrees applied on the gene HS.541791.

Fig. 3.4 presents the behavior of the quadratic (left) and linear (right) B-spline with different settings of the *knots* parameter. The oscillatory effect of the fitting curve is not a desired effect. The curve is forced to pass through the knots, thus affecting its shape. The order of the polynomials still dictates the degree of smoothness. As the number of knots is reduced, the fitting curve reflects a more natural behavior. Not being forced to pass through or get closer to specific points from the dataset, the B-spline basis provides an unbiased approach to the study. This is an effect that is desired, as all the samples forming the dataset are considered of equal significance.

The linear B-spline also reflects a behavior close to the desired effect, but as the number of knots is reduced, the result is very similar to the one provided by the linear regression models, which are not discussed in this paper.

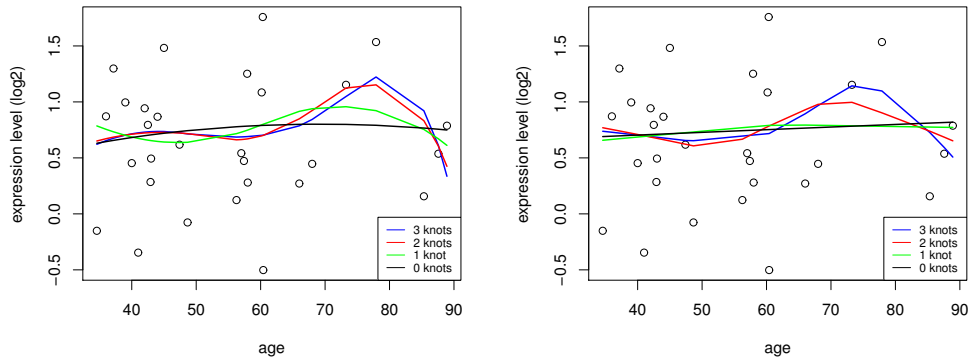


Figure 3.4.: Quadratic and linear B-spline with different settings of the *knots* parameter on the gene HS.541791.

From the above presented configurations of the B-splines, the most adequate for this study is the quadratic (e.g., *degree*=2) B-spline with no control points (e.g., *knots*=0). This configuration will provide an unbiased approach and allow the fitting curve the necessary freedom and smoothing degree. Furthermore, the "wiggle" effect is removed, thus the fitting curve behaves more natural when applied to the data. Another justification for this choice, and maybe the most important, is the possibility to easily customize the spline by changing its function basis. This allows the user to easily change the behavior of the fitting curve and/or adapt it to a specific dataset.
In the next section a different data smoothing method is presented and discussed. Following will be a comparison between the best candidates.

3.2. Non-linear local regression

Another data smoothing method is the local regression[7]. Locally weighted scatter-plot smoothing or LOESS (for short) is a locally weighted polynomial regression function. It is a "modern" data modeling method for data regression that uses the least squares regression[8].

[7] Local regression is a modeling method between a prediction variable and a response variable.
[8] A method for describing the relation between two variables by minimizing the sum of the squares of the errors computed between the observed value and the actual fitted value.

LOESS combines the simplicity of linear least squares regression with the flexibility of non linear regression. This is achieved by fitting simple models to localized subsets of the data in order to build up a function describing the deterministic part of the data variation, point by point. A variant of this smoothing function is LOWESS[19]. The main difference between the two is that the former uses local quadratic regression while the latter uses local linear regression. Moreover, LOWESS allows a weight vector to be applied on the data, thus each data point can be supplied with a weight value between 0 and 1. Both use direct evaluation at a reduced set of points, followed by interpolation. The interpolation algorithms are different [18]. LOESS uses quadratic interpolation, while LOWESS uses linear interpolation. For this study, the LOESS function was used.
The main reason why this function is preferred to LOWESS is that no parametrized fitness function has to be implemented, so no theoretical background of the model is needed. Furthermore, a weight vector is unnecessary as each data point is considered of equal importance in the context of an unbiased approach. A general drawback for the LOESS function is that it is computational intensive if applied on large datasets. As stated, the function uses quadratic interpolation as the quadratic B-spline, thus it is adequate for a comparison.
The syntax of the LOESS function in R[9] programming language is the following:

lo = loess(formula, span, degree, family)
predict(lo)

with *formula* specifying the numeric response and the numerical predictors. In this case only one predictor is used (e.g., age) and thus *formula* is set to $y\ x$ with x a vector containing the age values and y containing the gene expression level for each element of x. This type of formula is interpreted as a specification that the response y is modeled by a linear predictor specified symbolically by x. This model is a series of terms, each one being interpreted as the interaction of all the variables and factors that appear in that term.
The *span* parameter is the degree of the smoothing, similar to the *spar* parameter used in the spline function. It controls the size of the neighborhood used in the fitting, as the regression is done locally. The *degree* parameter represents the degree of the polynomials to be used in the interpolation. The possible values are 1 or 2 for linear or quadratic interpolation. As discussed in the previous section, the use of second-order polynomials is better suited, thus the

[9] R is a functional language built to statistically explore the data.

degree parameter is set to 2 for the rest of the results presented in this chapter.
The last parameter of the LOESS function is *family*. This parameter defines which fitting method will be used. The possible values are `gaussian` or `symmetric`. Using the former one, the fitting is done by the *least-squares* method, while the latter uses a re-descending M-estimator with the Tukey bi-weight function [17]. This method is more adequate in the presence of outliers in the dataset. As a PCA analysis was performed as described in Section 2 there is no need to take in consideration the hypothesis of existing outliers thus the `gaussian` setting was used for the *family* parameter.

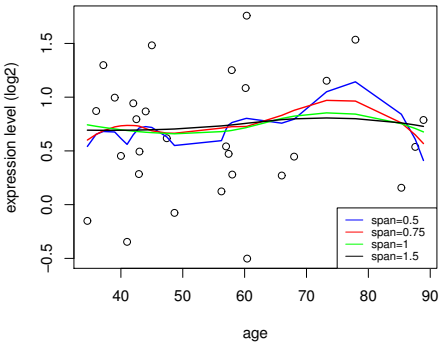


Figure 3.5.: LOESS function with different *span* settings applied on the gene HS.541791.

The return values of the LOESS function are passed through a predictor. The predictor is a generic function which looks in the data for explanatory variables to be used in the prediction of the values. In this particular case, the data is represented by the same x and y vectors forming a matrix.
Fig. 3.5 shows the results of the LOESS function with different settings for the *span* parameter. Similar to the *spar* parameter used in the splines, the *span* parameter brings the fitting curve to a more linear trend as increased. This behavior is due to the resize of the neighborhood that is used in the local regression. As the neighborhood size is increased, more data points are taken in consideration for the predictor, thus the fitting curve is not forced to bend so abruptly. For a *span* value that is ≥ 1 all the data points are used for the prediction of the curve.
Note that the kernel used in the LOESS function is tricubic weighting [20] thus taking the maximum neighborhood size (i.e., all the data points) respects the unbiased approach of the study. Still, due to the function kernel, a "wiggle" effect is noticeable in the shape of the fitting curve starting from the center of the neighborhood as third-degree polynomials are used in computing the weight of each point in the neighborhood. This weight is proportional to the distance as explained in [20]. For the rest of the results presented in this paper, the parameter *span* is set to 1.
The next section will present the best candidates among the data smoothing methods presented and will also provide a comparison in order to determine the best suited method applied to human gene expression.

3.3. Result evaluation

Out of the above described methods of data smoothing and their configuration, only a few presented the desired effect in the context of an unbiased study. From the *spline* family, the cubic natural spline and the quadratic B-spline are the best suited, as discussed in the previous sections. The LOESS function is taken in consideration as an alternative method to the spline functions and it is part of the comparison as it presents a desired smoothing effect. Fig. 3.6 depicts the effects of the chosen candidates on the LOC650826 gene (left) and on the MYH1 gene (right) which is a *known* probe from the dataset. By *known* is to be understood that it is a gene associated with muscular disorders and known for its down-regulation in expression during the lifetime of an individual. The LOC650826 is similar to the CG9804-PA in *Drosophila* [21] (i.e., fly). The choice of the MYH1 was made in order to present the effects of the candidate smoothing methods applied on a gene that is already known for its trend with time. The Bézier curves were also plotted as a reference for the two spline functions. The cubic spline has a *spar* parameter set to 1 and the basis of the B-spline was build using one linear term and one quadratic term. The *knots* parameter was set to NULL, thus the fitting curve of the quadratic B-spline has no control points. For the LOESS function, the *span* parameter was set to 1, thus the size of the neighborhood is equal to the number of data points of the probe. It can be considered equivalent in configuration with the quadratic B-spline as no control points are specified, but the center of the neighborhood still influences the shape of the fitting curve, as it can be seen in Fig. 3.6 (left) around age 60.

A statistical study was conducted in order to determine the degree of similarity between the smoothing methods that are considered the most appropriate to be applied on human gene expression. The parameter configuration used for each of the methods is the one presented in the previous sections. Table 3.1 summarizes the results of the statistics obtained. The correlation and covariance of two variables are statistical methods to measure the similarity degree of the variables involved. A correlation closer to 1 indicates that the two variables involved are highly related. The correlation is normally a value between $[-1, 1]$. A positive covariance indicates that the two variables change together, thus they show a relation. In the right column of the above table, the pairs of variables for which the statistics were computed are noted. The second column contains the average correlation (upper part of the table) and the average covariance values (lower part of the table) of the fitted values obtained from the entire dataset. In other words, the values were computed for the predicted

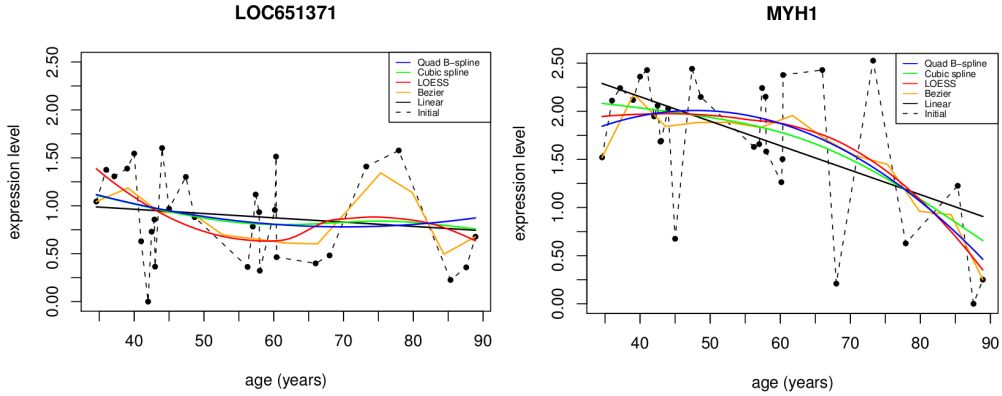


Figure 3.6.: Different methods applied to the gene LOC650826 (left) and to the MYH1 (right) gene.

Average correlation		
Comparison	Fitted	Residuals
Cubic spline - Quadratic B-spline	0.951	0.998
Quadratic B-spline - LOESS	0.885	0.994
LOESS - Cubic spline	0.965	0.996
Average covariance		
Comparison	Fitted	Residuals
Cubic spline - Quadratic B-spline	0.015	0.173
Quadratic B-spline - LOESS	0.017	0.171
LOESS - Cubic spline	0.016	0.170

Table 3.1.: The results of the statistical tests performed on the smoothing methods.

points (i.e., described curve) of the smoothing methods. The third column is similar to the second, except that the values were computed for the residuals of the smoothing methods. A residual value is the difference between the observed value and the fitted value. This difference in most of the cases is represented by the Euclidean distance between the two points or the sum of squares.

As a general conclusion of the above table, the three methods are very similar. The difference between the average correlation and average covariance values is to small to be considered insignificant, with respect to the dimensionality of the dataset. Note that these results were obtained from the entire dataset (i.e., $\approx 49,000$ probes), thus if referred to an individual probe, the statistical difference between the smoothing methods is hardly noticeable or even inexistent.

Another study on the behavior of these smoothing methods was conducted: *cross-validation*. "Leave-one-out" cross-validation was applied in order to see which method behaves better at predicting the data. The idea behind this type of cross-validation is to leave one data point out at a time and see how well the function can predict the missing point. In this case, the cross-validation was performed for each of the 29 data points (i.e., 29 K-fold).

Cross-validation	
Function	Value
Cubic spline	0.209
Quadratic B-spline	0.224
LOESS	0.194

Figure 3.7.: Cross-validation performed on the smoothing methods.

Fig. 3.7 contains the results of the "leave-one-out" cross-validation. The right column holds the values of the *average sum of squares* which is an error measurement method. It computes the average squared distance between the fitting curves as one data point is left out at a time in order to see how well it is predicted. Note that the values are a result of the leave-one-out cross-validation applied on the entire dataset (i.e., $\approx 49,000$ probes). The first conclusion to be drawn is that the average sum of squares is almost the same for the three smoothing methods, which denotes their robustness. If computed per probe, the average value of the sum of squares is very small. A second conclusion is that the three methods behave very similar even if their internal mechanisms differ considerably thus the configuration of the parameters is correct. All candidates present the desired smoothing effect when applied on the data and show a reliable degree of robustness.

This statistical study shows that the cubic natural spline, the quadratic B-spline and the LOESS function, with the described parameter configuration are highly similar and provide the desired data smoothing effect. The results of the "leave-one-out" cross-validation denote that the methods are robust. The cubic natural spline behaves as expected when its smoothing parameter has a high enough value, but control points are still taken in consideration for the fitting curve, thus not allowing the user to completely influence the behavior of the spline function.

The LOESS function also presents the desired smoothing effect with the right setting of its smoothing parameter. Taking a maximum neighborhood size (i.e., all the data points of a probe) respects the unbiased approach of this study, but the center of the neighborhood still influences the shape of the predicted curve (seen as a control point) due to the internal mechanism of the function as presented above.

The quadratic B-spline is the most adequate data smoothing method in the context of an unbiased study of human gene expression. The fact that it is

highly configurable gives the user the possibility to model the fitting curve and adapt it to each dataset. As depicted, even with the simplest parameter configuration (e.g., no knots) it provides the desired smoothing results. For this study, no control points were defined as all the data points are considered of equal significance. Different behaviors of the fitting curve can be obtained with simple manipulations of the spline basis and furthermore, the data distribution or representation does not yield any special computational cases to be taken in consideration.
This aspect is one of the most important for this study, as the datasets used are partly composed of samples originating from individuals having the same age. This raises a problem for other smoothing methods and the details (e.g. birth date of the individuals) allowing to compute the exact age until the sampling and eliminate the duplicate abscissa values are not always at hand. Therefore, for the rest of the results presented in this paper, the quadratic B-spline was used as the data smoothing method.

3.4. Data filtering

Data smoothing is a very important step in data mining and pattern recognition as the variation between the data points is present, especially in the type of data used in this study. After the preprocessing steps described in Chapter 1, which bring the dataset to a more standardized representation, reduced noise and eventual outliers discarded, data smoothing brings the extremes in the dataset to the mean, thus refining the data for the following steps. In order to discard the "uninteresting" probes and reduce the size of the dataset, a filtering step is necessary before further analysis. This is the next step in the data processing as depicted in the flow diagram in Appendix A.
For this study, the filtering is based on the age associated p-value of each probe, obtained after applying ANOVA[10] on two linear regression models fitted for each of the probes. A typical linear regression model has the form $response \sim terms$ where *response* is the numeric response vector and *terms* is a series of terms which specifies a linear predictor for response. In this case, the *terms* are represented by the age vector (i.e., the age of the individuals for each of the samples) and *response* is the vector containing the expression level associated to each element of the age vector.

[10] Analysis of variance (ANOVA) is a statistical test built to indicate if the means of two populations are equal. It indicates if the variability within and between the populations is significantly different.

Tissue	Probes	Sign. probes	Percentage
quadriceps	48803	4101	8.40
brain frontal cortex	12558	2339	18.62
kidney cortex	22283	2397	10.75
kidney medulla	22283	1474	6.61

Table 3.2.: The number of significant probes for each dataset and the percentage their represent from the initial number of probes.

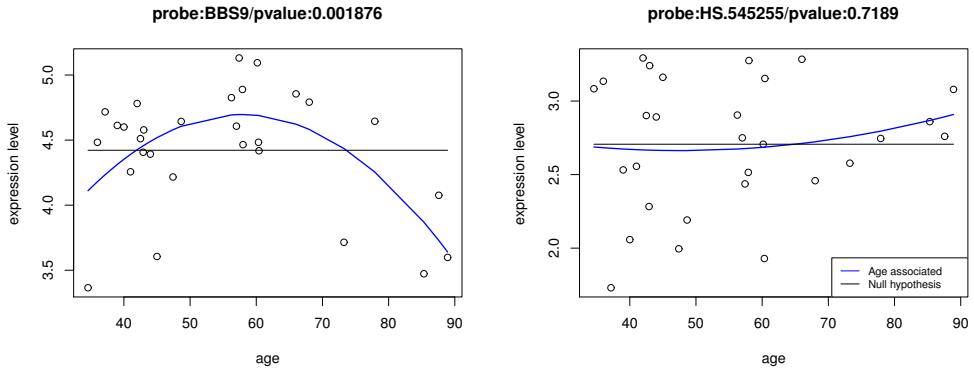


Figure 3.8.: An age-associated significant probe (left) versus an insignificant one (right).

One of the two linear regression models needed is the data smoothing method used (e.g., the quadratic B-spline). The splinebasis represents the linear predictor for the response vector. To compute a p-value relevant to the analysis, the second linear regression model to be fitted and passed to ANOVA in order to test the significance of the linear regression, assumes there is no age associated effect (i.e., the linear predictor for the response vector is set to 1). This is commonly known as the null hypothesis[11] and in this case it was built by not taking in consideration the age of the individuals (i.e., assuming the changes in the expression levels are not due to the age of the individuals). For more details regarding linear models and ANOVA the reader is referred to [10][11]. A probe is considered to be *significant* if the ANOVA results in a p-value<0.05 [9]. All the probes having a p-value above this threshold are considered *insignificant*, thus discarded from the dataset. Table 3.2 contains a summary of the number of significant probes after the filtering for each of the datasets. Note that even for the kidney medulla tissue, which resulted in an insignificant p-value after

[11] The null hypothesis is usually the hypothesis that sample observations result purely from chance.

the global test, as pointed in Fig. 2.1, there were still $\approx 1,500$ significant probes identified, but the amount of change in the gene expression levels is not as remarkable as it is in the other tissues as it will be shown in the next chapter. The last column in the table contains the percentage of data remaining after the filtering for each dataset.
Fig. 3.8 depicts two random probes from the quadriceps dataset during the filtering step. The difference between the two is obvious. The left probe presents a much more stringent curve relative to the null hypothesis (e.g., the black line) than the one on the right. This comparison reflects the age associated effect on the samples. In other words, the associated p-value of each probe reflects in one way the amount of change in expression level a probe is subject to along with the age. This can also be visually confirmed by looking at the y axis of the plots and quantifying the difference between the change in expression level of the two probes.
Data smoothing and filtering refined the datasets and reduced their dimensionality, thus preparing them and facilitating the next step of the analysis.

4. Data clustering

In order to identify the major trends in a dataset, the data has to be clustered (i.e., grouped) by similarities. This is one of the most important tasks in data mining and statistical analysis. The clustering criteria (e.g., low distance, conceptual clustering, etc.,) are mostly user defined as data clustering is more of task than a specific algorithm. It is considered to be an unsupervised learning[12] problem, as no values or clues denoting an a priori grouping of the data are given in general. In other words, it is a process of organizing data into groups, or clusters so that the members of a cluster are similar in some way but dissimilar to the members of another cluster. There are numerous clustering algorithms that can be applied on a dataset, but one has to find the best suited to be applied depending on the expected results.
For this study, two major data clustering concepts were applied on the data. They are presented in detail in the next sections.

4.1. Hierarchical clustering

Connectivity based clustering or hierarchical clustering is a statistical method based on the idea that objects (e.g., observations or probes in this particular case) are more related to neighboring objects than to objects farther away. It is intended to find relatively homogeneous clusters depending on the measured characteristics. These clustering algorithms connect objects based on their distance to form clusters. The clustering process either starts with each object in a separate cluster and then sequentially merges the clusters, reducing their number at each iteration until only one cluster is left, either starts with one cluster and successively "subclusters" until each object is in one cluster. Namely, a cluster is described by the maximum distance required to connect the parts of its subclusters. This process can be represented as a dendrogram, or tree, therefore the name hierarchical clustering, where each iteration in the process

[12] Unsupervised learning, as part of machine learning, studies how systems can learn to represent input data such that it reflects the overall statistical structure of the input data.

is depicted by a connection of the branches in the tree. The advantage of this type of clustering is the fact that the user does not need to specify a number of desired clusters. But as it is shown in the following sections, this advantage can turn into a burden. Two major types of hierarchical clustering exist: *agglomerative* and *divisive*. Both are detailed below, as well as their results when applied to the quadriceps muscle dataset.

Agglomerative

Bottom-up hierarchical clustering algorithms treat each observation as a cluster and then successively merge pairs of clusters until only one cluster is formed containing all the probes. The bottom-up algorithms are also known as *agglomerative* algorithms due to the fact that pairs of clusters are combined at each iteration, thus agglomerated. The distance metric used in the agglomerative clustering was the Euclidean distance, as it is one of the most commonly used for hierarchical clustering and adequate for this study due to its unbiased nature. A dissimilarity matrix[13] is computed from the initial set of observations using the Euclidean distance as metric in order to facilitate the hierarchical clustering. The connection criterion for two clusters is average-linkage or unweighted pair group method with arithmetic mean (UPMGA for short) which computes the distance between two clusters as the average pair-wise distance between the members of two clusters. There are several other linkage methods for hierarchical clustering, but for the purpose of this study, average-linkage is chosen in order to minimize the within cluster variation. For the interested reader, more details about the algorithm are to be found in Chapter 5 of [24].

Divisive

Divise or top-down hierarchical clustering algorithms start by putting all the observations (i.e., probes) in the dataset in one cluster and then recursively divide it, thus the name *divisive*. At each iteration of the algorithm, the cluster with the largest dissimilarity between any two of its observations is chosen. This dissimilarity is computed with the help of a dissimilarity matrix, as for the agglomerative clustering, and the Euclidean distance as metric. In order to divide the selected cluster, the algorithm first looks for the largest average dissimilarity to the other observations in the selected cluster. A new cluster

[13] Dissimilarity matrix or distance matrix describes the pair-wise distinction (distance) between the elements of the matrix by associating a score to each element.

is formed by this observation and in the subsequent steps of the algorithm, observations that are closer to the one chosen are associated to the new cluster. This results into a division of the selected cluster into two new clusters. The steps are repeated until each cluster is formed by a single observation. For the algorithm details, one is referred to Chapter 6 of [24].

Comparison

A result of the agglomerative hierarchical clustering applied on the significant probes from the quadriceps dataset is depicted in Appendix B. The *agnes* function from the R package *cluster* [30] was used with the aforementioned parameter configuration. Note that a dissimilarity matrix of the input data may be used instead of the one computed internally by the function. The bottom part of the dendrogram (i.e., hierarchical tree) represents the early phase of the clustering algorithm, with each probe defining a cluster and successively merging them until only one cluster remains, as it can be observed in the upper part of the dendrogram. The length of a branch in the hierarchical tree denotes the strength of a cluster. The longer the branch, the more robust the cluster is. The choice of the cut-off point (i.e., decide upon the number of remaining clusters) proved to be a problem. The clearer part of the plot (e.g., starting around value 10 on the y axis) indicates that there are 4 clusters that can be considered robust enough for further analysis, but the far right part of the plot shows that a smaller cluster is preserved in the tree until the latest step in the clustering algorithm. The length of the branch emerging from this cluster is an indication that one is intended to keep the cluster, thus the cut-off point in the tree has to be lowered (e.g., around value 5 on the y axis) which implies the presence of clusters that are more unstable. The choice of the cutting point is different in each dataset and it is somewhat arbitrary.
Appendix C presents the result of divisive hierarchical clustering applied on the quadriceps dataset. The *diana* function of the *cluster* R package was used with the specified configuration. As for the *agnes* function, the input of a dissimilarity matrix is optional. The upper part of the dendrogram is the early phase of the clustering algorithm, starting from one cluster containing all the probes in the dataset and successively dividing it into two subclusters at each iteration. As for the agglomerative hierarchical clustering, the first choice of a cut-off point of the hierarchical tree would be around the value 4 on the y axis, but the length of the branches indicates that the clusters are not that robust. Climbing down the tree towards longer branches (e.g., around value 3 on the y axis) suggests the presence of stronger clusters in the left side of the plot, but it

becomes blurry on the left side. This is subject to change in the other datasets used in the study. Once again, the choice of the cut-off point represents a problem by not being able to provide a clear distinction between stable clusters and unstable. This is a known problem of the hierarchical clustering and several methods were put in place in order to determine the number of clusters in a dataset [22][23] but not implemented in this study as another type of clustering was chosen in order to overcome this issue.

4.2. *K*-means clustering

K-means clustering is probably the mostly used clustering algorithm, next to hierarchical clustering applied in data mining and statistical analysis. It aims at partitioning the observations from a dataset into k clusters such that the sum of squares from the data points of an observation to the center of the cluster is minimized. The partitioning of the observations results into a Voronoi tessellation (or diagram) which aims at dividing a space into subsets (or cells). The subsetting is determined by a distance measure (e.g, Euclidean distance or sum-of-squares) to other subsets in the given space. Each object (e.g., observation or probe) is associated with a Voronoi cell such that the distance from the center of the cell to the object is less than the distance between the object to any other cell (or subset) in the space. At the minimum, all cluster centers are at the mean of their Voronoi cell. This is a special case of Voronoi tessellation called centroidal, where the center of a Voronoi cell coincides with its center of gravity.
This particular case of tessellation represents the base of several algorithms used in the k-means clustering method. The most commonly used algorithms are MacQueen [25] and Hartigan-Wong [26] which will always return the desired number of clusters. If an initial matrix of centers (i.e., cluster centroids) is provided, it may occur that no observation is close enough to one or more centers, thus the resulting number of clusters may differ from the initial k. In this situation, the Lloyd [27] and Forgy [28] algorithm is better suited, but for the purpose of an unbiased study (i.e., without any assumption or prior knowledge of the data) as this, the Hartigan-Wong algorithm was used as no predefined cluster centroids were computed. This is the preferred k-means clustering algorithm when using Euclidean distance as metric, as it outperforms the others in terms of minimization of the within cluster variation (i.e., sum-of-squares of the data points to the cluster center).

Classic *K*-means

The general k-means clustering algorithm is a heuristic algorithm[15] therefore it is not expected to return the global optimum solution at every run. The algorithm may get stuck in a local optima, thus several random starts are advised. An outline of the algorithm can be found in Appendix D.
A number of maximum iterations can easily be implemented if the algorithm fails to converge and becomes time consuming, but the probability of retrieving a local solution rather than the global one is much higher. The number of desired clusters (e.g., k parameter) is the tricky part of the algorithm and the major problem. It performs better when a higher number of clusters are expected as it has more freedom and comparison terms, thus the accuracy of assigning an observation to the appropriate cluster is higher. As the number of clusters decreases, the chances of getting stuck in a local optima become higher. Of course, this is subject to the size and quality of the dataset. Several methods for estimating the number of clusters have been proposed. A comprehensive comparison of more than 30 methods can be found in [23]. Depending on the probability distribution of the dataset, one method can outperform another, but may produce worse results in a different setting. Overall, the choice of k is proven to be very difficult, especially applied on microarray data [29] in the context of gene clustering. This is due partly to the complexity of the underlying biological processes that take place in an organism and dictate the gene interactions.
Considering the size of the datasets involved in this study, as detailed in Table 2.1 the choice of a reasonable number of clusters can become a major problem. The preceding steps in the general workflow prepared the data to allow cleaner clustering and reduced considerably the dimensionality of the datasets. Nevertheless, the estimation of k is difficult, as the within cluster variation is also to be kept at a minimal value. Of course, the two are correlated: the smaller the number of clusters, the bigger the variation within a cluster. In order to minimize the variation and avoid the local optima solution, a variant of the k-means algorithm was applied.

[15] Heuristic algorithms are often used in optimization problems. They iteratively find solutions by seeking through the entire search space.

Genetic *K*-means

Genetic algorithms are a class of evolutionary algorithms[16] that uses natural models in the context of heuristic search for optimization problems based on one or more fitness functions. The general steps of a canonical genetic algorithm (or GA for short) are the following:

choose random initial population
evaluate fitness of initial population
repeat
  select best individuals
  apply crossover
  apply mutation
  evaluate fitness
until end condition

At each iteration of the algorithm, a new generation of solutions is created. The population representing this generation is selected based on various criteria such as elitist selection (i.e., the survival of the fittest). The fitness functions are specific to each GA depending on the purpose of the algorithm. The *crossover* and *mutation* probabilities are user defined in most of the cases, thus one of the operators can easily be inhibited. Ending conditions are to be implemented with respect to the aim of the optimization and/or the resources (e.g., fixed number of generations, fitness threshold). For more details regarding genetic algorithms, one is referred to [31].
For this study, the optimization problem is the minimization of the total within cluster variation (i.e., sum-of-squares error), thus the fitness function is a distance-based function. As for the classic k-means algorithm, the metric is the Euclidean distance. An outline of the genetic k-means applied on the datasets can be found in Appendix E. Note that the *crossover* operator is missing as it is not applicable in this context. An implementation of the algorithm, developed to be a fast genetic k-means algorithm (or FGKA for short) [32] was adapted for this study. It is an interpretation of the originally developed genetic k-means algorithm (GKA) [33]. An incremental version of the genetic k-means algorithm (IGKA) with applications on gene expression data is also implemented by the same authors as FGKA, but the results yielded did not differ substantially from FGKA in the context of this study, thus the latter is used for its simplicity and superior speed. The genetic algorithm is assumed to provide the global opti-

[16] Evolutionary algorithms use nature inspired models such as mutation, crossover, selection in order to solve optimization problems by simulated evolution.

Genetic K-means				
mp	4 clusters	8 clusters	12 clusters	24 clusters
0.01	**1318.41**	**708.31**	838.78	826.95
0.02	1318.42	710.96	**493.22**	864.54
0.04	1318.72	748.42	503.75	820.77
0.05	1318.71	714.70	1017.33	870.84
0.06	1319.29	727.00	495.87	822.52
0.10	1319.58	728.61	880.68	901.44
0.15	1322.80	748.42	598.20	**778.74**
0.20	1320.18	788.09	879.17	807.26
0.50	1326.82	840.20	997.53	824.73
0.75	1338.44	940.80	845.18	906.68

Table 4.1.: The resulting total within cluster variation after applying FGKA on the quadriceps muscle dataset.

mum solution at each run as stated in [32] with the right configuration of the specific operators (e.g., mutation probability). For the results presented in the next section, the FGKA was applied on the quadriceps muscle dataset, against the *kmeans* function from the R package *stats* using the Hartigan-Wong algorithm and Euclidean distance as metric. More details regarding the FGKA and the operators used in the algorithm are to be found in [32]. Both FGKA and IGKA are available upon demand near the authors.

Comparison

The two k-means algorithms presented above were applied on the smoothened and filtered quadriceps muscle dataset. The 4101 significant probes remaining were grouped in 4, 8, 12 and 24 clusters respectively using Euclidean distance as metric. The resulting total within cluster variation (i.e., sum of the within variation over all clusters) obtained was summarized for each of the two algorithms. Table 4.1 shows the results of the FGKA. In bold font, the best results are highlighted for each number of clusters. Note that for the genetic k-means, the best value for the mutation probability (e.g., first column in the table) had to be identified in order to obtain the best result for a specific number of clusters. For each k present in the table, the algorithm was run for 150 generations with a solution population size of 50. In most of the cases, it converged before 100 generations and the average time per iteration was ≈ 1.5 seconds. Higher values used for the population size did not influence the final total within cluster

Classic K-means				
run	4 clusters	8 clusters	12 clusters	24 clusters
1	**1318.40**	708.40	503.08	271.53
2	1352.43	708.92	491.30	270.14
3	**1318.40**	717.76	492.47	**270.13**
4	**1318.40**	720.92	497.02	271.94
5	1345.55	**708.21**	499.95	271.66
6	**1318.40**	721.61	494.77	270.04
7	**1318.40**	788.21	497.03	274.20
8	**1318.40**	716.61	493.40	277.37
9	1352.43	710.48	**491.01**	270.61
10	**1318.40**	717.8	495.47	275.62

Table 4.2.: The resulting total within cluster variation after applying *kmeans* on the quadriceps muscle dataset.

variation (TWCV for short), only the speed of the algorithm. Table 4.2 summarizes the results obtained using the *kmeans* function from R base package *stats*. For each number of clusters, the algorithm was run 10 times in order to see the impact of the random initialization of the cluster centers in the first steps of the algorithm. As the table shows, the best results obtained are overall better than the ones presented by FGKA. Moreover, the *kmeans* function used also performed better in terms of speed, compared to FGKA, as the algorithm converged almost instantaneous. This is due most probably to the dimension of the dataset as the input data used in testing the FGKA presented a lower dimensionality.
Appendix F and Appendix G depict the 8 clusters obtained with classic k-means and genetic k-means respectively applied on the quadriceps muscle dataset. Note the similarity between the two resulting plots. In the title of each plot, the number of probes associated, as well as the within sum-of-squares (e.g., WSS) for each cluster are denoted. The within sum-of-squares is the within cluster variation. From the point of view of results, the algorithms are comparable. Due to the fact that FGKA has to be adapted (i.e., mutation probability, number of generations) for each number of desired clusters and for each dataset, the classic k-means algorithm implemented in the R package *stats* was chosen over this implementation of genetic k-means. Not to mention the fact that it outperforms FGKA in computation time.

4.3. Distance metric

The decision upon the number of clusters remains a problem. The less the number of clusters and less within variation, the better. As Appendix F shows, even if all the significant probes are grouped in only 8 clusters, the results are robust, as denoted by the 95 and 99 percentiles[17] (e.g., blue and black lines) in the cluster plot. This represents, to some extent, the confidence interval of a cluster, as the majority of the probes associated to a specific cluster (i.e., falling under the 95 percentile) denote the amount of variation within the cluster. As the number of clusters increases, the two percentiles become closer to the cluster center. Moreover, one can easily see that in general, each trend of the data identified by the cluster centroids has its symmetric by the x axis (e.g., age). A representative example are clusters 1 and 6 or 2 and 7 in the Appendix F. This phenomenon was identified in all the datasets available for this study and after a visual analysis it was concluded that for a desired number of clusters bigger than 4, each identified trend has its symmetric by the x axis.
As it can be seen in Appendix F the two main categories of identified trends are the probes that present an increase in the gene expression level (i.e., y axis value) and the ones presenting a decrease respectively. The probes that are subject to define a linear, non-progressive curve were eliminated during the filtering step presented in the previous chapter. The purpose of the study being to identify genome-wide expression trends and keeping in mind that for this step, the increase or decrease in expression level does not yield individual consideration, another distance metric was used in order to reduce even further the number of clusters. To take advantage of the symmetry of the clusters, the absolute correlation[18] was used as a distance measure. This measure allows the probes that present a symmetric trend to be grouped together, thus the number of clusters can be reduced by a factor of at least two, depending on the dataset. The classic k-means clustering algorithm was applied on the significant probes of each dataset using absolute correlation as a distance measure and the obtained results, as well as their interpretation are presented in the next section.

[17] A percentile shows the distribution of the values in a set of data divided in 100 subsets. The percentile of a particular value represents the percentage of the values smaller than the specified value.

[18] Absolute correlation distance uses the absolute value of the correlation between the observations within a set of data.

4.4. Result evaluation

The function *Kmeans* from the R package *amap* [34] was used with the *method* parameter set to `abscorrelation`. This parameter indicates which distance metric to be used. The function is assumed to produce the same results as the one from the *cluster* package using the Euclidean distance. Internally, the same k-means clustering algorithms are implemented, but the choice of distance metrics is much more extended in the k-means function of the *amap* package. In order to establish the number of clusters, a visual analysis was conducted. Appendix H shows the results of k-means clustering with absolute correlation as a distance measure, applied on the significant probes from the quadriceps muscle dataset. As one can easily see, reducing the number of clusters from 4 to 2 does not yield important increase in the within cluster variation, as denoted by the 95 and 99 percentiles. Moreover, two clusters (e.g., Cluster 1 and Cluster 2) are proven to be robust when the data is grouped in 3 and 4 clusters respectively. If grouped in 4 clusters, the two most robust clusters are found back and a third one is formed as a combination of the two (e.g., Cluster 3), thus decreasing k to 2 is the intuitive proceeding action. This phenomenon is equally present in the other datasets.
Furthermore, using the absolute correlation as a distance metric for clustering, provided some useful visual representation of the points in time (i.e., on the age scale) where the gene expression is subject to change. This point is defined by the intersection of the symmetric trends that are clustered together. Informally, the points where the genes start to vary (e.g., increase or decrease in the gene expression) were called *switching points*. Graphically, it is the part of the plot with the smallest variation (i.e., where the percentiles come the closest to the cluster centroids). In most of the datasets used, two switching points were identified and defined as the *early* and *late* switching points, even though from a biological point of view, a more appropriate definition would be the *early variable genes* and *late variable genes*. For the rest of the paper, the notation of switching points will be used. The table below summarizes the results of the absolute correlation clustering applied on the available datasets. Appendix I illustrates the above table. Each cluster obtained with absolute correlation was split into two subclusters using the *kmeans* function from R with Euclidean distance as metric, in order to quantify the probes that present an upregulation or downregulation[19]. Analyzing the two subclusters obtained with Euclidean distance, one can easily observe how the switching points are found in the ab-

[19] Upregulation or downregulation is the process by which a gene increases or decreases in expression due to increased or decreased transcription of a specific mRNA.

Dataset details		1^{st} switching point				2^{nd} switching point			
Tissue	# probes	age	# probes	upreg.	downreg.	age	# probes	upreg.	downreg.
quadriceps A	4101	42-47	2716	1241	1475	70-80	1385	499	886
quadriceps B	6448	38-43	2451	1386	1065	58-68	3997	1396	2601
brain frontal cortex	2339	50-55	1149	638	511	75-80	1190	392	798
kidney cortex	2397	60-70	1729	392	854	65-75	668	335	333

Table 4.3.: The identified switching points in gene expression and their associated probes.

solute correlation clusters.

In order to verify the robustness of the clustering, as well as the influence of the number of samples in the dataset upon the obtained clusters, several validation tests were put in place. Note that in Table 4.4 a second quadriceps muscle dataset is present (e.g., quadriceps B). This dataset was used only for validation purposes, as the quality of the data was not sufficiently high to make it part of the study. The age range of the samples in the quadriceps B dataset is from 17 to 89. Some of the samples are the same as in quadriceps A, as the data generated was present on multiple batches. Quadriceps B represents the merging of two batches, while quadriceps A is the data generated from only one batch.

Other validation tests were performed on the available datasets. A variant of the kidney cortex dataset was obtained by discarding every other sample from the original dataset. This test was conducted in order to observe the impact of the dataset resolution upon the obtained clusters. Fig. 4.1 depicts the clusters obtained with absolute correlation on the original dataset (above) and the aforementioned variant (below). Note that the main difference between the clusters obtained from the two datasets is to be found in the number of significant probes. The amount of within cluster variation is more increased in the half of the dataset, due to the decreased resolution of the dataset. Appendix J displays the two clusters obtained with absolute correlation as distance metric on the two quadriceps muscle datasets. One can easily see the bigger variation for the quadriceps B clusters. The two switching points in the gene expression level are preserved in both datasets. A shift on the x axis on the switching point occurrence is noticeable. This behavior is due to the sample distribution of the dataset and the smoothing method used (i.e., the freedom of the curve). In average, for the tissues available in this study, the shift on the age position was identified to be between 3 and 5 years. Appendix K illustrates the shifting of the switching points, as the age range of the brain frontal cortex dataset is shortened by ≈ 10 years at each try. Note that the late switching point is preserved, regardless the change in age range, as opposed to the early switching point, which fades away due to the lack of samples. In other words, the first switching point in the brain dataset, was identified between the age of 50 and

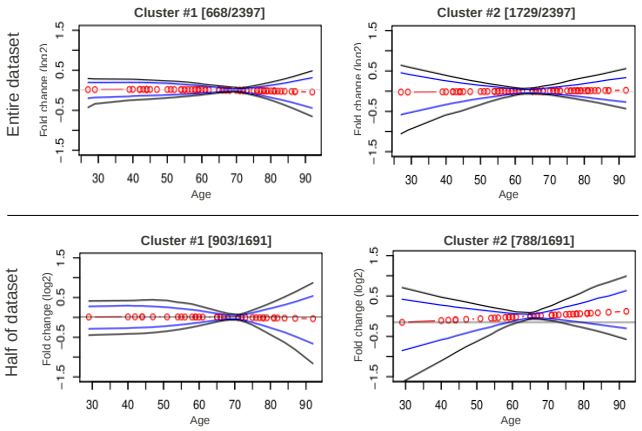


Figure 4.1.: The absolute correlation clusters obtained from the entire kidney cortex dataset and half of the dataset.

Dataset details		Early switching point		Late switching point	
Tissue [samples]	age range	age	# probes	age	# probes
quadriceps A [29]	35-89	42-47	2716	70-80	1385
quadriceps B [25]	17-89	38-43	2451	58-68	3997
brain frontal cortex [30]	26-106	50-55	1149	75-80	1190
brain frontal cortex [25]	36-106	53-58	534	78-83	1320
kidney cortex [72]	27-92	60-70	1729	65-75	668
kidney cortex half [36]	27-92	60-70	788	65-75	903
kidney medulla [61]	29-92	60-75	691	NA	NA

Table 4.4.: The identified switching points in gene expression during the validation tests.

55, as denoted in Table 4.4. Increasing the starting age in the dataset is equivalent to removing samples from the left side of the x axis, thus finally decreasing the resolution to only 2 samples for the first switching point, as depicted in Appendix K. The shifting of the late switching point is also noticeable, as mentioned, but translated into biological time, it may be considered insignificant. The clustering of the kidney cortex dataset yielded two individual switching points, as Fig. 4.1 (above) depicts and Table 4.4 quantifies. The important age overlap between the two denotes the presence of only one switching point, as a merge of the initial two.

As a conclusion, the resolution of the datasets does not influence the occurrence of the switching points in gene expression, but the age range has an impact (e.g., shifting) on their positioning (i.e., x axis), as Table 4.4 summarizes. The clustering of the data always yields results. Even when applied on the kidney medulla dataset, which did not pass the global test, the absolute correlation k-means clustering managed to group the data, as Fig. 4.2 shows.

Note that the amount of change in gene expression level (e.g., y axis) is not

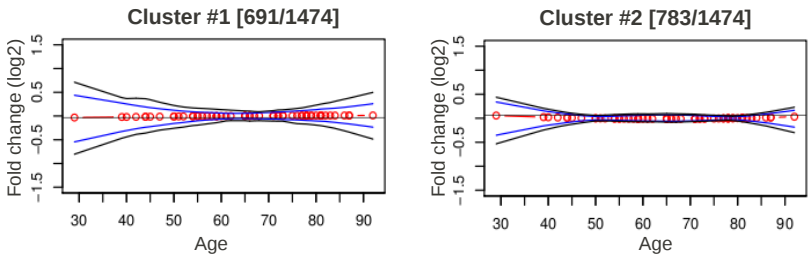


Figure 4.2.: The absolute correlation clusters obtained from the kidney medulla dataset.

Original dataset				Permuted dataset		
Tissue	samples	age range	# probes	avg. # probes	std. dev	SP
quadriceps	29	35-89	4101	2291	861	93
brain frontal cortex	30	26-106	2339	677	522	97
kidney cortex	72	27-92	2397	1136	772	89
Kidney medulla	61	29-92	1474	1090	700	76

Table 4.5.: The resulting significant probes from the shuffled samples of the datasets after 100 permutations.

as significant as in the other tissues. This represents also a validation of the global test. Keeping in mind the fact that the data will always be clustered, thus patterns will be identified regardless the representation of the data, another validation test was conducted. The samples of the available datasets were shuffled (i.e., permuted) and the general workflow of processing applied from step one on the resulting datasets.

Table 4.5 summarizes the number of age associated probes (i.e., significant probes) after 100 random permutations of each dataset. Note that the average number of significant probes is considerably low in the shuffled datasets, as the fifth column of the table denotes. The next column contains the standard deviation of the number of significant probes found at each permutation of the samples. The last column (e.g., SP) shows the percentage of sample permutations that resulted in a smaller number of significant probes than the original dataset. Appendix L depicts the results of clustering on the permuted samples of the brain frontal cortex dataset and kidney cortex dataset. The shape of the clusters, as well as the position of the switching points in the shuffled datasets does not correspond with the ones in the original datasets, which proves the consistency of the age associated filtering and clustering of the probes.

The various validation tests conducted on the datasets and their variants conclude that the smoothing method implemented, as well as the age-associated filtering of the probes are consistent to yield robust clusters obtained with absolute correlation as distance metric for the k-means clustering. The subclustering using Euclidean distance for the k-means validates the existence of symmetric trends in the gene expression profiles. Therefore, the resulting clusters are ready for the next step in the general workflow, detailed in the next chapter.

5. Biological functionality

In order to reconnect with the biological part of the study, an analysis on the functionality of the identified clusters was performed. Note that the study described in this paper was performed on skeletal muscle tissue (e.g., quadriceps muscle) and the aforementioned and detailed steps in the processing of a gene expression dataset were validated on the other available tissues. This step is crucial, as it represents itself a validation of the entire process. In other words, an analysis on the biological function of the resulting clusters from the quadriceps tissue, should yield muscle associated molecular and biological processes. Moreover, if the hypothesis is right, ageing-associated differences between the early and the late switching point should be revealed. The following sections describes in detail this analysis and its results, as well as its validation on the brain frontal cortex tissue.

5.1. Genome annotation

The first step is to identify which genes are present in the resulting clusters. Note that the clusters obtained with absolute correlation as a distance metric were used, as the upregulation or downregulation of the genes does not require a distinct analysis at this point. The point in time (i.e., age) where the variation of the genes occurs is more important than its trend. As stated in the first part of this study, the quadriceps dataset is generated from biopsies of healthy individuals, each sample containing $\approx 49,000$ probes. Depending on the instruments and the protocol used in the data generation, the probe annotation (i.e., the target set of transcripts of a given gene) differs. For the quadriceps tissue, the technologies developed by Illumina[20] were applied in the DNA sequencing, as for the validation dataset (e.g., brain frontal cortex), the Affymetrix[21] tools were used. A comparison of the two can be found in [35] as it does not represent the purpose of this paper. Subsequently, the annotated probes are to

[20] http://www.illumina.com/
[21] http://www.affymetrix.com/

Dataset details		1^{st} switching point				2^{nd} switching point			
Tissue	# probes	age	# probes	upreg.	downreg.	age	# probes	upreg.	downreg.
quadriceps A	3094	42-47	2000	990	1010	70-80	1094	409	685
brain frontal cortex	2150	50-55	1021	583	438	75-80	1129	339	790
kidney cortex	1851	60-70	1851	911	940	NA	NA	NA	NA

Table 5.1.: The resulting unique genes with an Entrez Gene annotation.

be mapped to their homologous genes. This is done via a genome annotation[22] database. There are numerous such databases (e.g., Ensembl, UniProt, RefSeq), aiming at gathering complete genome information regarding protein encoding in organisms. The Entrez Gene[23] database was used for this study, as it focuses on the genomes that have been completely sequenced and it represents the result of many other databases intensively maintained and updated through the National Center for Biotechnology Information (or NCBI for short).
The consistency of the different genome annotations varies and it is a known problem in the field of bioinformatics. Not all genes are identified or annotated in the genome, thus the coverage of the mapping differs. Note that a gene is expressed twice in average, thus duplicates are also present among the genes forming one cluster. The Table 5.1 summarizes the results of probe mapping to their respective Entrez Gene Id and removed duplicates. Note that the quadriceps A dataset was used (i.e., age range 35-89). The Entrez annotation covered $\approx 75\%$ of the initial number of significant probes in the quadriceps dataset and $\approx 92\%$ in the brain dataset. This discrepancy is mainly due to the aforementioned different DNA sequencing methods used in the process. The generation of the kidney cortex dataset made use of the same method as for the brain (e.g., Affymetrix). Regardless, the Entrez Gene annotation mapped only $\approx 77\%$ of the probes, as opposed to the more extensive coverage present in brain. Note that only one switching point is present in Table 5.1 for the kidney dataset, as discussed in the previous chapter.
The duplicates were removed by cluster, as one gene can be present in both clusters due to its number of occurrences. The resulting lists of genes were mapped individually to the GO database[24] in order to obtain detailed, reliable and consistent [36] biological information about the involved genes. This was achieved using the *org.Hs.eg.db* database from the Bioconductor [37] package in R language. Other tools for retrieving gene ontology vocabulary terms (or GO terms for short) are also available (e.g., DAVID) and provide (among others)

[22] Genome annotation is the process of identifying the gene locations, their coding regions in a genome and describe their functionality.
[23] `http://www.ncbi.nlm.nih.gov/sites/entrez?db=gene`
[24] The Gene Ontology (or GO) is a bioinformatics project aiming at standardizing the gene representation and their biological attributes via a controlled vocabulary of terms.

the functionality of gene conversion between different genome annotations. The *org.Hs.eg.db* is preferred due to the control over the database version and simplicity of the output. The resulting lists of GO terms contained a huge amount of redundancies, thus they were subject to an analysis in order to determine their enrichment[25] among the rest of the genes present in the genome.

5.2. Filtering

The lists of GO terms per cluster were filtered by several criteria. First of all, a measure of their enrichment was conducted. This was achieved using Fisher's exact test[26] on the lists of genes. A contingency table[27] necessary for the exact test was generated for each of the GO terms mapped to an Entrez Gene annotation. More in detail, for each GO term, the 2 by 2 matrix representing the contingency table summarizes its number of occurrences in the cluster as opposed to the number of occurrences in the rest of the mapped genes and the occurrence number of the rest of the mapped GO terms in the cluster as opposed to the number of their occurrence among the rest of the mapped genes. Thus, the null hypothesis being that the GO term has no particular association with the cluster that is part of. For the interested reader, details about Fisher's exact test and constructing contingency tables can be found in [38].
The resulting p-value of the exact test represents the primary filtering criterion. Only the GO terms associated with a p-value < 0.05 were considered significant [9], thus preserved. The p-values were a priori corrected for multiple comparisons, using the false discovery rate control (or FDR for short) in order to adjust for the amount of false positives resulted by incorrect rejection of null hypotheses during the Fisher's exact test. The threshold for FDR is computed based on the distribution of the p-values. For more details regarding the FDR control, one is referred to [39]. The second filtering threshold was based on the number of associated genes to a specific GO term. This number reflects the level of generality of the GO term. The more generic the term is, the more associated genes. Keeping in mind that this study aims at identifying specific age associated biological processes, the generic terms do not present any substantial interest. On the other hand, if a GO term has to little associated genes, it may

[25] The enrichment of a GO term is a statistical measure to determine whether the observed level of annotation for a group of genes is significant in a background set of genes.

[26] Fisher's exact test is a statistical significance test used to determine if the associations between two categorical variables are not the result of random.

[27] Contingency tables are a format used to analyze the frequency distribution of categorical variables.

indicate that its contribution is not relevant enough. Thus, the threshold for the associated genes of a GO term was fixed somewhat arbitrarily not to be higher than 1,000 and less than 10.
The next filtering criterion was to remove the redundant GO terms within a cluster. This redundancy is expressed by two GO terms having the same associated genes and the same age-regulated genes (i.e., genes present in the input cluster) associated. The phenomenon occurs due to the interrelated GO terms. Once the redundancies removed, the resulting GO terms lists were manually filtered in order to remove the entries that are not related to the study (i.e., organ development and other irrelevant biological processes) and thus prepare them for the next step.

5.3. Hierarchical representation

The structure of the gene ontology database is a directed acyclic graph[28] (or DAG for short), thus it facilitates a hierarchical representation of the identified significant GO terms. Nevertheless, the DAG structure has several consequences. The most important, related to this study, is that one GO term can have multiple parents in the hierarchical tree, thus redundancies are still to be expected, even after the thorough filtering steps. Hierarchical trees of the GO terms associated to each cluster were generated using recursive SQL[29] queries on the GO database in order to determine the relationship between the GO terms within a cluster. The query returns the immediate neighbors (e.g., one level up or one level down in the DAG) of a GO term which are to be found among the ones present in the input cluster. The recursive call of the SQL queries for each of the GO terms in a cluster allows to form a well defined hierarchy between the members of the same group.
Appendix M and N shows the most important clusters of GO terms associated with age regulated genes and their biological classification in the quadriceps dataset, for the early and late switching point respectively. Note that expected molecular and biological processes that are to be found in skeletal muscle tissue are present in both switching points (e.g., muscle contraction cluster). This denotes the consistency of the methodology described in the previous chapters. The hierarchical trees formed were clustered by their biological functionality,

[28] A directed acyclic graph is a directed graph structure in which the vertices are connected to one another, but there is no path to follow starting from one vertice and ending up in the same vertice.
[29] Structured Query Language or SQL is a database specific programming language.

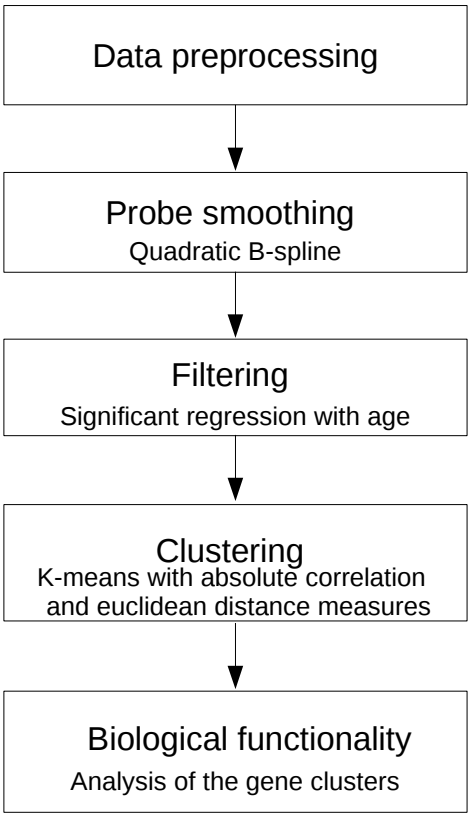
in order to give more information regarding the complexity of the involved processes. Some of the GO terms clusters that were identified in the brain frontal cortex tissue are listed in Appendix O and P. The number of significant GO terms was considerably higher, along with the redundant terms. This is due to the different probe annotation system and the discrepancies between the gene annotation tools. As a consequence, the complete list of identified GO terms was not presented due to its size. Again, note the existence of tissue specific significant GO terms which validates once again the analysis.
This representation will serve as a good basis for further understanding of the complex molecular mechanisms associated with ageing, as well as a comparison between the early and late switching points.

6. Conclusion

The purpose of this study is to give an insight of the underlying molecular processes associated with the physiological ageing of humans by studying the genome-wide expression trends. The datasets generated from healthy tissue samples were subject to preliminary processing steps in order to bring the data to a more standardized representation and diminish the influence of the different instruments and protocols used in the microarray generation. Several data smoothing methods were tested and evaluated on the skeletal muscle (i.e., quadriceps) dataset, as detailed in Chapter 3, in order to determine the best suited for gene expression. The smoothing of the data represents an essential step that allows to reduce inter-individual variation by bringing the extremes in the data to their mean. Moreover, it allows a better representation for the following steps in the process. The probes that did not present a significant age-association were discarded during the filtering step and the resulting ones grouped by their trend using k-means clustering algorithm. The different distance metrics used in the clustering step and their performance on the available datasets, as well as their evaluation are detailed in Chapter 4. The biological functionality of the identified gene clusters in the quadriceps tissue and frontal cortex of the brain was studied using the Gene Ontology database and their associated significant biological and molecular processes clustered and presented under a hierarchical form in order to understand the root cause of the variation in gene expression. The hierarchical trees proved to be consistent with respect to the source of the samples. This paper described a step-by-step methodology to follow in genome-wide expression trend analysis, regardless the topic of the research. Gene expression analysis was formerly performed, but not genome-wide. In this particular study, the methodology yielded the expected results, thus validating the interpretation and use of the above described steps for microarray dataset processing.

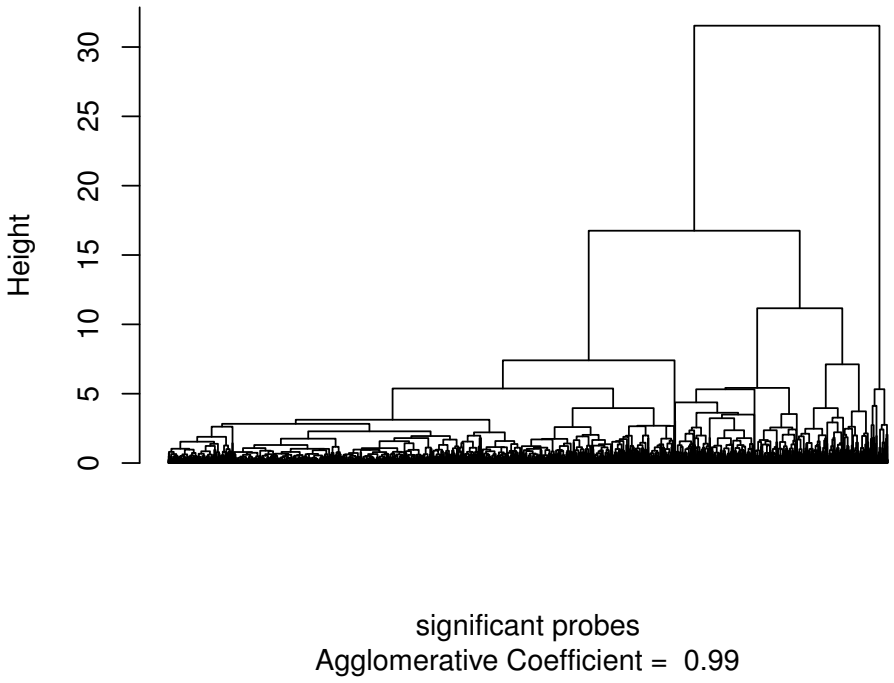
A. Flow diagram

The steps of the general workflow for processing a microarray dataset.



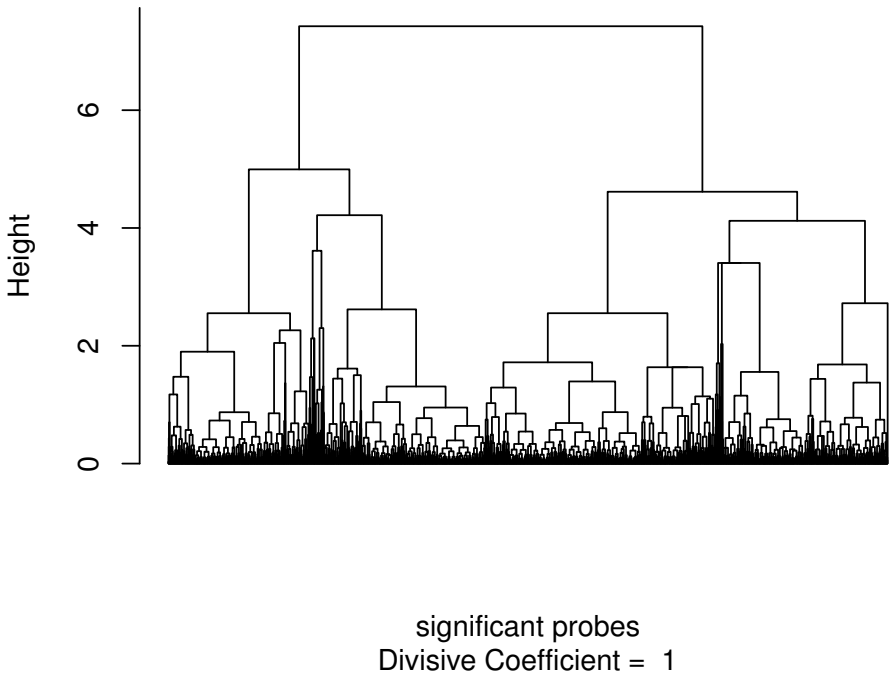
B. Agglomerative hierarchical clustering

The results of the *agnes* function from the R package *cluster* applied on the significant probes from the quadriceps muscle dataset. Note the robustness of the far right cluster which suggests that the cut-off point should be around value 5 on the y axis.



C. Divisive hierarchical clustering

The results of the *diana* function from the R package *cluster* applied on the significant probes from the quadriceps muscle dataset. The shape of the dendrogram does not indicate a suitable cut-off point as the clusters are not robust enough relative to the length between two divisions.



D. *K*-means clustering algorithm

The pseudocode for the classic k-means clustering:

`input:`
$O \leftarrow o_1, .., o_n$ (the observations to be clustered)
k (the number of desired clusters)
`output:`
$C \leftarrow c_1, .., c_k$ (the cluster centroids)
$m : O \rightarrow C$ (the membership of an observation to a cluster)

function $kmeans(O, k)$
for $c \in C$ **do**
  $c_i \leftarrow o_x \in O$ (random observation)
end for
for $o_i \in O$ **do**
  $m(o_i) \leftarrow argmin_{n \in 1,..,k}(distance(o_i, c_n))$
end for
while $isChanged$ **do**
  for $c_i \in C$ **do**
    $update(c_i)$
  end for
  for $o_i \in O$ **do**
    $minDist \leftarrow argmin_{n \in 1,..,k}(distance(o_i, c_n))$
    if $minDist \neq m(o_i)$ **then**
      $m(o_i) \leftarrow minDist$
      $isChanged(m) \leftarrow TRUE$
    else
      $isChanged(m) \leftarrow FALSE$
    end if
  end for
end while
return C, m

E. Genetic *K*-means clustering algorithm

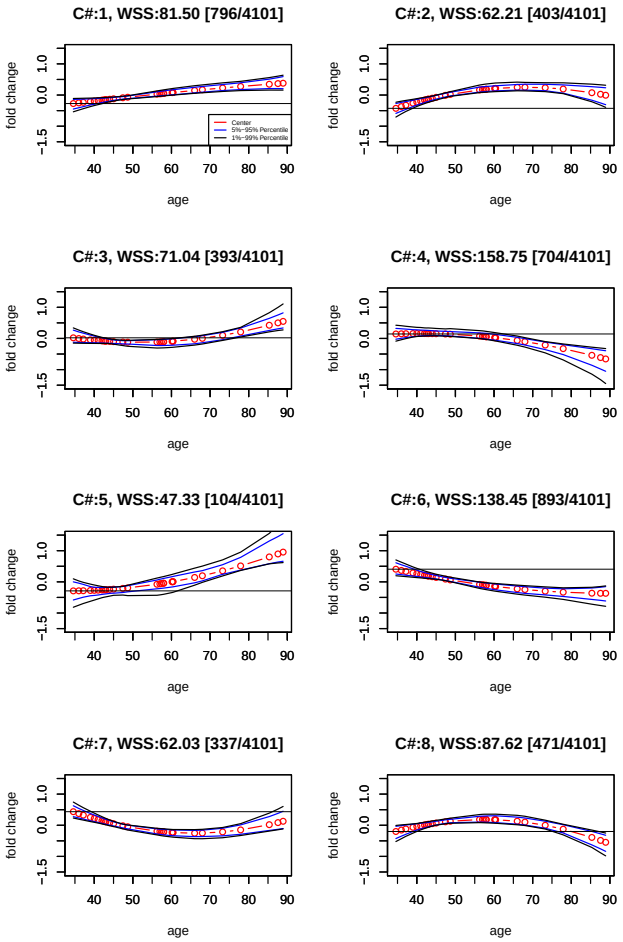
The pseudocode for the genetic k-means clustering. Note the use of the evolutionary operators. The *kmeans* operator incorporates the algorithm described in the previous appendix.

```
input:
O ← o_1, .., o_n (the observations to be clustered)
k (the number of desired clusters)
mp (the mutation probability)
G (the number of generations)
S (the size of the solution population)
output:
P ← c_1, .., c_k (the solution population including membership)

function kmeans(O, k, mp, G, S)
TWCV ← MAX (set the total within cluster variation to maximum)
P ← initialize(O, k, S) (random initialization of solution population)
TWCV ← evaluate(P)
while i ≤ G do do
  P'_i ← select(P_i) (select best individuals)
  P''_i ← mutate(P'_i, mp)
  P''_i ← kmeans(P''_i, O, k) (apply classic k-means)
  TWCV_i ← evaluate(P''_i)
  if TWCV_i ≤ TWCV then
    TWCV ← TWCV_i
    P_{i+1} ← P''_i
  end if
  i ← i + 1
end while
return P
```

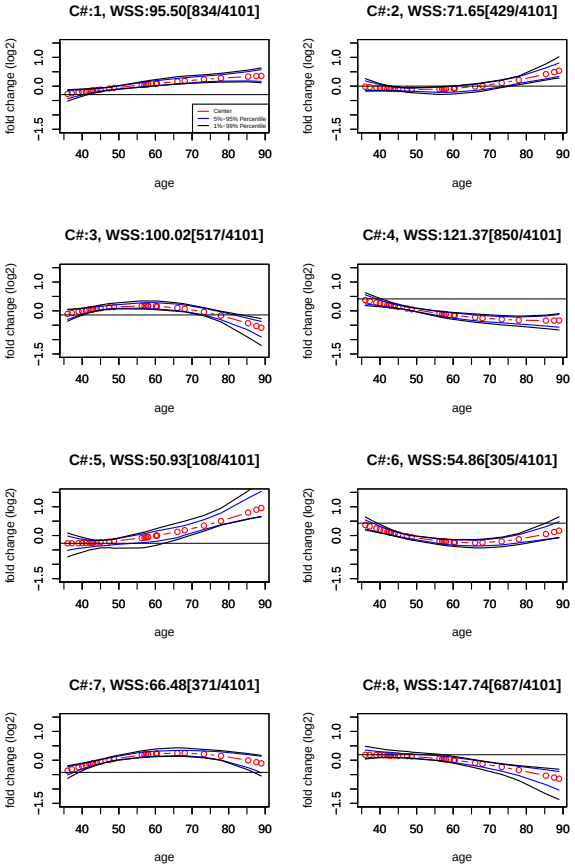
F. Classic *K*-means clustering

Resulting clusters obtained with the *kmeans* function from R base packages applied on the significant probes of the quadriceps muscle dataset. The Euclidean distance was used as a metric and k is set to 8 (i.e., number of clusters). Note the horizontal symmetry of most of the trends described by the cluster centroids. The 95 and 99 percentiles denote the within cluster variation, quantified in the title of each plot as the within sum of squares (WSS) value.



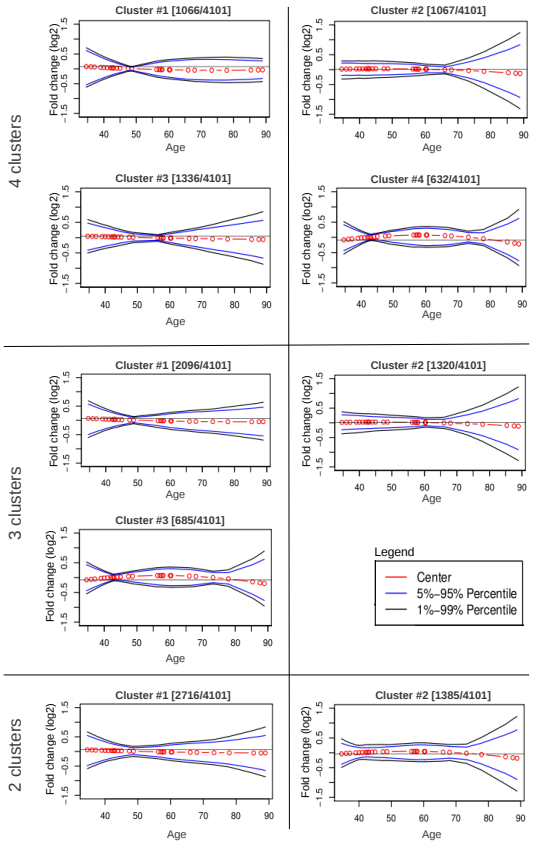
G. Genetic K-means clustering

Resulting clusters obtained with the FGKA algorithm applied on the significant probes of the quadriceps muscle dataset. The Euclidean distance was used as a metric and k is set to 8 (i.e., number of clusters). Note that the horizontal symmetry of the cluster centroids is also present as for the classic k-means and moreover, it shows the consistency of the clusters obtained with the classic algorithm, as a validation. The 95 and 99 percentiles denote the within cluster variation, quantified in the title of each plot as the within sum of squares (WSS) value. The results of the two k-means algorithms are comparable, with respect to the shape, within cluster variation and number of probes associated to each cluster.



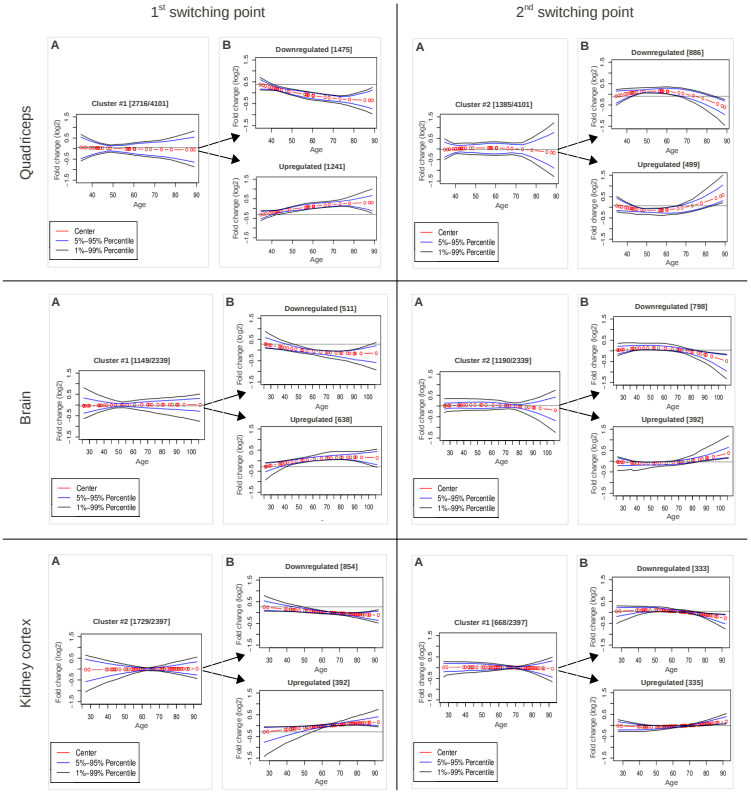
H. *K*-means clustering with absolute correlation as distance metric

The significant probes from the quadriceps dataset were grouped in 4, 3 and 2 clusters respectively, using the *Kmeans* function from the R package *amap* with absolute correlation as distance metric. Note the consistency of the first two clusters throughout the process, therefore the choice of only two clusters to define all significant probes.



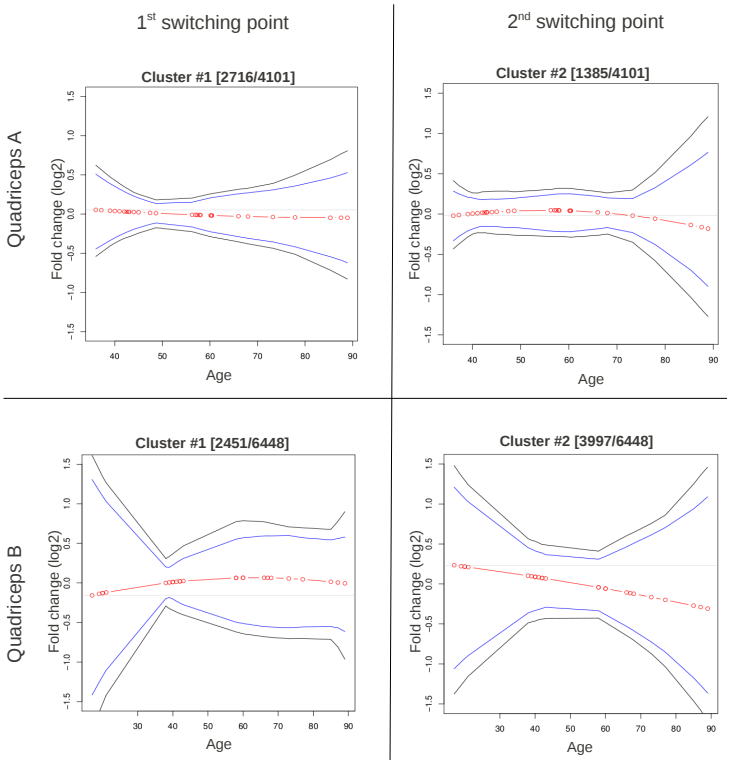
I. The identified switching points

In each of the three tissues, the significant probes were grouped into two clusters using absolute correlation (A) and each cluster divided into two subclusters using Euclidean distance (B) as metric, in order to validate the horizontal symmetry of the trends and quantify the number of upregulated and downregulated probes. Note that the cluster centroids reflect more the distribution of the upregulated and downregulated probes, not their trend. This is visually defined by the percentiles.



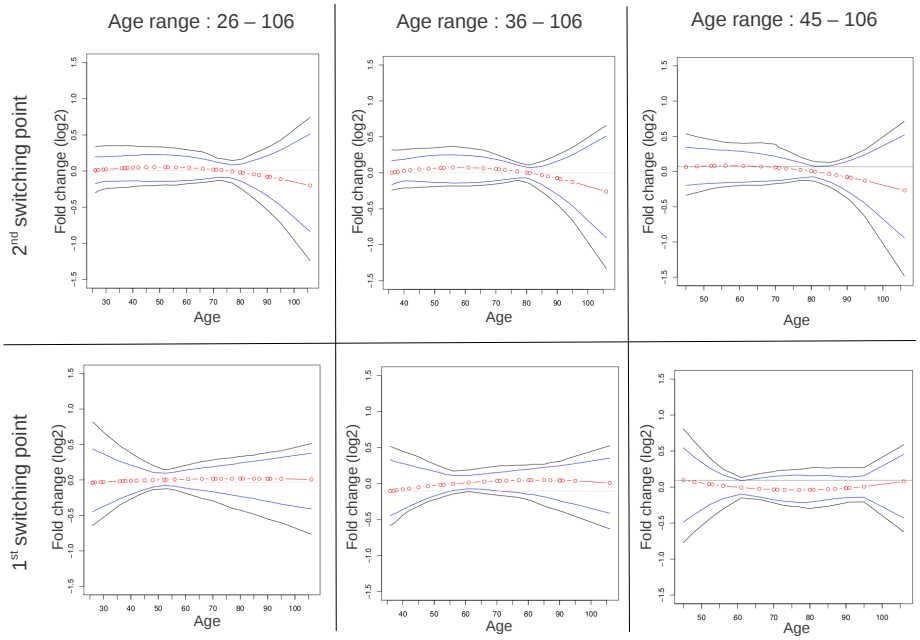
J. Switching points in quadriceps

A secondary quadriceps dataset (e.g., Quadriceps B) was used for the validation of the identified switching points. This dataset was obtained by merging two batches and not used in the actual analysis due to its poor quality, denoted by the considerably higher within cluster variation. Note that the age range of the Quadriceps B (17–89) dataset differs from Quadriceps A (35–89) thus the occurence of the switching point is influenced.



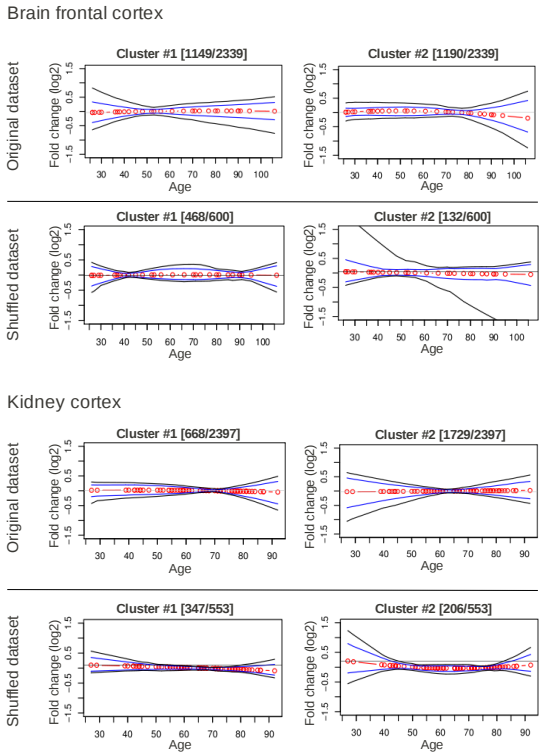
K. The robustness of the late switching point in brain tissue

Two variants of the original brain frontal cortex dataset were created by decreasing the age range (i.e., removing samples from the left side) in order to verify the robustness of the identified switching points. Each of the variants decreases the age range by approximately a decade. Note that the late switching point is preserved, as opposed to the early, which fades due to the lack of resolution.



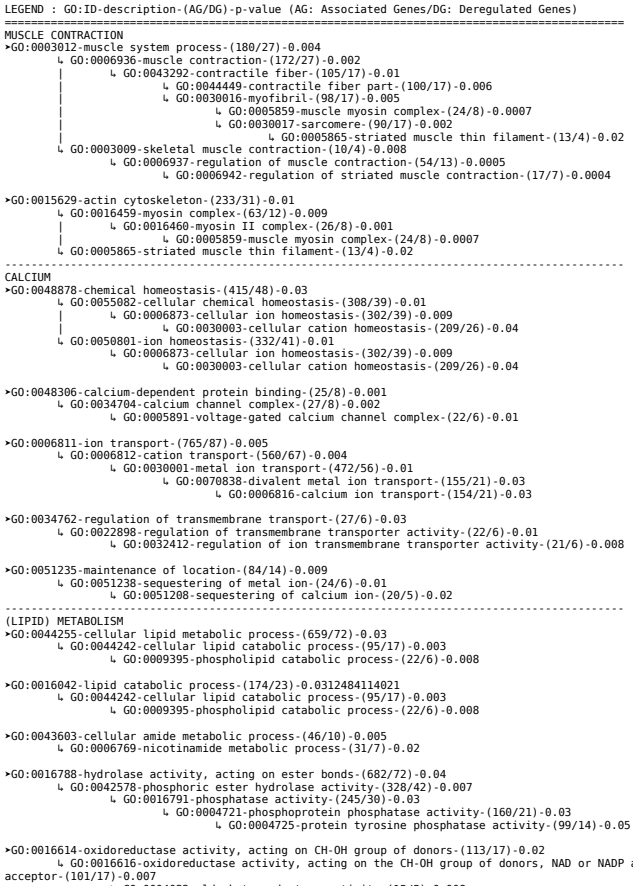
L. Clusters of the permuted samples

In order to verify that the obtained clusters do not represent an artifact of the data smoothing and clustering algorithms, the samples of the datasets were permuted (i.e., shuffled) and the processing protocol applied from the first step. One can easily see that the shape of the clusters resulting from the shuffled datasets differs from the original dataset.



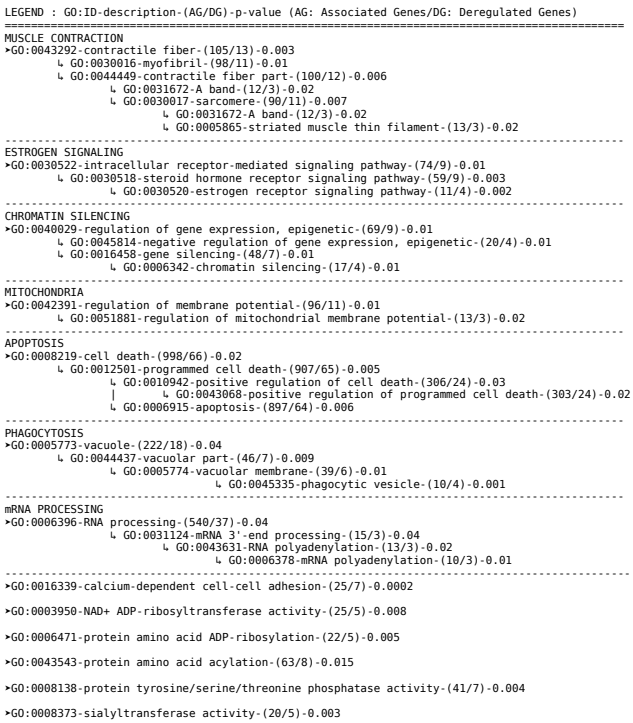
M. GO term clusters in the early switching point in the quadriceps

The major clusters of the significant GO terms identified in the early switching point in quadriceps under hierarchical representation based on their ontology. Note the specificity of the terms with respect to the tissue of origin.



N. GO term clusters in the late switching point in the quadriceps

The major clusters of the significant GO terms identified in the late switching point in quadriceps under hierarchical representation based on their ontology. Note the specificity of the terms with respect to the tissue of origin and the molecular processes associated with late age (e.g., apoptosis, chromatin silencing).



O. Some GO term clusters in the early switching point in the brain

A part of the major clusters of the significant GO terms identified in the early switching point in the brain frontal cortex under hierarchical representation based on their ontology. The terms are specific with respect to the tissue. The entire list was not attached due to its considerable size.

```
LEGEND : GO:ID-description-(AG/DG)-p-value-(AG: Associated Genes/DG: Deregulated Genes)
=============================================================================================
NERVOUS SYSTEM
➤GO:0007399-nervous system development-(837/90)-8.2e-07
        ↳ GO:0007417-central nervous system development-(314/37)-0.001
        |       ↳ GO:0007420-brain development-(198/24)-0.01
        |       ↳ GO:0048709-oligodendrocyte differentiation-(18/5)-0.04
        ↳ GO:0007272-ensheathment of neurons-(35/8)-0.02
        ↳ GO:0022008-neurogenesis-(398/43)-0.002
                ↳ GO:0048699-generation of neurons-(373/39)-0.008
                        ↳ GO:0030182-neuron differentiation-(335/37)-0.005
                                ↳ GO:0048666-neuron development-(242/26)-0.04
                                        ↳ GO:0031175-neuron projection development-(189/23)-0.02

➤GO:0007610-behavior-(380/42)-0.002
        ↳ GO:0007611-learning or memory-(65/12)-0.01
                ↳ GO:0007613-memory-(18/6)-0.01

➤GO:0000267-cell fraction-(830/76)-0.001
        ↳ GO:0005626-insoluble fraction-(652/66)-0.0003
                ↳ GO:0005624-membrane fraction-(636/63)-0.0009
                        ↳ GO:0019717-synaptosome-(50/10)-0.01

➤GO:0007267-cell-cell signaling-(575/67)-3.5e-06
        ↳ GO:0007268-synaptic transmission-(280/41)-3.8e-06
---------------------------------------------------------------------------------------------
ACTIN CYTOSKELETON
➤GO:0015629-actin cytoskeleton-(233/27)-0.01
        ↳ GO:0005884-actin filament-(29/8)-0.008

➤GO:0051693-actin filament capping-(22/6)-0.03

➤GO:0008092-cytoskeletal protein binding-(459/53)-6.3e-05
        ↳ GO:0003779-actin binding-(310/34)-0.007

➤GO:0007010-cytoskeleton organization-(419/55)-7.2e-07
        ↳ GO:0030036-actin cytoskeleton organization-(239/33)-0.00015
        |       ↳ GO:0007015-actin filament organization-(110/17)-0.005
        ↳ GO:0000226-microtubule cytoskeleton organization-(123/16)-0.04
        ↳ GO:0051494-negative regulation of cytoskeleton organization-(55/13)-0.0004
                ↳ GO:0031111-negative regulation of microtubule polymerization
                      or depolymerization-(16/6)-0.008
                        ↳ GO:0007026-negative regulation of microtubule depolymerization-(15/6)-0.005

➤GO:0022411-cellular component disassembly-(110/17)-0.008
        ↳ GO:0032984-macromolecular complex disassembly-(87/16)-0.002
                ↳ GO:0034623-cellular macromolecular complex disassembly-(68/16)-0.0001
                        ↳ GO:0043624-cellular protein complex disassembly-(61/16)-3e-05
                                ↳ GO:0051261-protein depolymerization-(43/12)-0.0003
                                        ↳ GO:0007019-microtubule depolymerization-(16/6)-0.008
                                                ↳ GO:0007026-negative regulation of microtubule
                                                                depolymerization-(15/6)-0.005
---------------------------------------------------------------------------------------------------------
CELL CYCLE
➤GO:0007049-cell cycle-(842/77)-0.001
                ↳ GO:0022403-cell cycle phase-(422/43)-0.007
                        ↳ GO:0051325-interphase-(104/15)-0.02
                                ↳ GO:0051329-interphase of mitotic cell cycle-(98/15)-0.01
                                        ↳ GO:0000082-G1/S transition of mitotic cell cycle-(46/10)-0.008
---------------------------------------------------------------------------------------------------------
APOPTOSIS
➤GO:0008219-cell death-(998/87)-0.002
        ↳ GO:0010941-regulation of cell death-(645/59)-0.01
                ↳ GO:0043067-regulation of programmed cell death-(643/59)-0.009
                        ↳ GO:0043069-negative regulation of programmed cell death-(297/37)-0.0006
---------------------------------------------------------------------------------------------------------
CELL MIGRATION
➤GO:0016477-cell migration-(303/31)-0.043
        ↳ GO:0030334-regulation of cell migration-(126/19)-0.005
                ↳ GO:0030335-positive regulation of cell migration-(60/12)-0.006
```

P. Some GO term clusters in the late switching point in the brain

A part of the major clusters of the significant GO terms identified in the late switching point in the brain frontal cortex under hierarchical representation based on their ontology. The terms are specific with respect to the tissue and late age molecular pathways are present, as in the quadriceps. The entire list was not attached due to its considerable size.

```
LEGEND : GO:ID-description-(AG/DG)-p-value-(AG: Associated Genes/DG: Deregulated Genes)
================================================================================================================
NERVE CELL SYSTEM
➤GO:0007399-nervous system development-(837/90)-4.1e-06
        ↳ GO:0022008-neurogenesis-(398/45)-0.001
        |       ↳ GO:0048699-generation of neurons-(373/41)-0.005
        |       |       ↳ GO:0030182-neuron differentiation-(335/36)-0.01
        |       |               ↳ GO:0048666-neuron development-(242/27)-0.04
        |       |                       ↳ GO:0048667-cell morphogenesis involved in neuron
                                                differentiation-(166/22)-0.01
        |       |                       |       ↳ GO:0007409-axonogenesis-(152/21)-0.01
        |       |                       ↳ GO:0031175-neuron projection development-(189/23)-0.03
        |       |                               ↳ GO:0048812-neuron projection morphogenesis-(164/22)-0.01
        |       |                                       ↳ GO:0007409-axonogenesis-(152/21)-0.01
        |       ↳ GO:0042063-gliogenesis-(51/10)-0.02123859
        ↳ GO:0048468-cell development-(610/65)-0.0002
                ↳ GO:0042995-cell projection-(476/51)-0.002
                        ↳ GO:0043005-neuron projection-(175/27)-0.0004
                        |       ↳ GO:0030424-axon-(64/12)-0.01
                        ↳ GO:0043025-cell soma-(73/13)-0.01

➤GO:0007268-synaptic transmission-(280/47)-4.3e-08
        ↳ GO:0050804-regulation of synaptic transmission-(66/13)-0.005
                ↳ GO:0048167-regulation of synaptic plasticity-(29/9)-0.002
                ↳ GO:0007270-nerve-nerve synaptic transmission-(32/12)-1.7e-05

➤GO:0045202-synapse-(279/49)-2.8e-09
        ↳ GO:0044456-synapse part-(191/29)-0.0002
                ↳ GO:0008021-synaptic vesicle-(55/11)-0.015
----------------------------------------------------------------------------------------------------------------
APOPTOSIS
➤GO:0008219-cell death-(998/112)-1.3e-08
        ↳ GO:0012501-programmed cell death-(907/96)-2.8e-06
                                ↳ GO:0051402-neuron apoptosis-(57/11)-0.02
                                        ↳ GO:0043523-regulation of neuron apoptosis-(50/11)-0.007
----------------------------------------------------------------------------------------------------------------
NUCLEOTIDE BIOSYNTHESIS
➤GO:0009165-nucleotide biosynthetic process-(261/40)-3.6e-06
        ↳ GO:0009141-nucleoside triphosphate metabolic process-(120/27)-1.8e-07
                ↳ GO:0009144-purine nucleoside triphosphate metabolic process-(113/25)-7.9e-07
                        ↳ GO:0009205-purine ribonucleoside triphosphate metabolic process-(112/25)-6.9e-07
                                ↳ GO:0046034-ATP metabolic process-(100/22)-5.1e-06
                                        ↳ GO:0006754-ATP biosynthetic process-(89/21)-2.9e-06
----------------------------------------------------------------------------------------------------------------
ENERGY BIOGENESIS
➤GO:0016818-hydrolase activity, acting on acid anhydrides, in phosphorus-containing
        anhydrides-(722/81)-2.1e-06
        ↳ GO:0016462-pyrophosphatase activity-(719/81)-1.9e-06
                ↳ GO:0017111-nucleoside-triphosphatase activity-(690/81)-3.8e-07
                        ↳ GO:0016887-ATPase activity-(317/40)-0.0003
                                ↳ GO:0042623-ATPase activity, coupled-(252/33)-0.001
                                |       ↳ GO:0042626-ATPase activity, coupled to transmembrane movement
                                |               of substances-(104/20)-0.0002
                                |               ↳ GO:0042625-ATPase activity, coupled to transmembrane
                                |                       movement of ions-(69/18)-8.1e-06
                                |                       ↳ GO:0015662-ATPase activity, coupled to
                                |                               transmembranemovement of ions,
                                |                               phosphorylative mechanism-(49/12)-0.001
                                |                       ↳ GO:0019829-cation-transporting ATPase
                                |                               activity-(32/10)-0.0005
                                |                               ↳ GO:0046961-proton-transporting ATPase
                                |                                 activity, rotational mechanism-(20/ 6)-0.02
                                ↳ GO:0003924-GTPase activity-(191/27)-0.001

➤GO:0006091-generation of precursor metabolites and energy-(324/40)-0.0002
        ↳ GO:0015980-energy derivation by oxidation of organic compounds-(151/18)-0.04
        ↳ GO:0006119-oxidative phosphorylation-(99/16)-0.001
```

Bibliography

[1] Microarray Analysis and Gene Expression Profiling, `http://www.microarrayworld.com/`, accessed on December 11th, 2011.

[2] B. Brais, *Oculopharyngeal muscular dystrophy: a late-onset polyalanine disease*, Cytogenet Genome Res 2003, Vol. 100, pg[252–260] (DOI: 10.1159/000072861).

[3] B. Udd, *Distal muscular dystrophies*, Handbook of clinical neurology, 2011, Vol. 101, pg[239–262].

[4] A. Chartier, V. Raz et al.,*Prevention of oculopharyngeal muscular dystrophy by muscular expression of Llama single-chain intrabodies in vivo*, Human Molecular Genetics, 2009, Vol. 18(10), pg[1849–1859], DOI: 10.1093/hmg/ddp101.

[5] H. Kawai, *Miyoshi distal muscular dystrophy (Miyoshi myopathy)*, Brain nerve, 2011, Vol. 63(2), pg[147–56].

[6] A. Brazma et al., *Minimum information about a microarray experiment (MIAME)-toward standards for microarray data*, Nature Genetics, 2001, Vol. 29, Part 4, pg[365–372], Nature Publishing Co., ISSN 1061–4036.

[7] W.E. Johnson, A. Rabinovic and C. Li, *Adjusting batch effects in microarray expression data using Empirical Bayes methods*, Biostat (2007) Vol. 8, pg[118–127], doi: 10.1093/biostatistics/kxj037.

[8] J.J. Goeman et al., *A global test for groups of genes: testing association with a clinical outcome*, Bioinformatics 2004 Vol. 20 (1): pg[93–99], DOI: 10.1093/bioinformatics/btg382

[9] R.A. Fisher, *Statistical Methods and Scientific Inference*, 1956, New York: Hafner.

[10] H.F. Senter, *Applied linear statistical models*, Journal of the American Statistical Association, 2008, Vol. 103, Iss. 482.

[11] R.A. Fisher, *The use of multiple measurements in taxonomic problems*, Annals of Human Genetics, 1936, Vol. 7, pg[179–188], DOI: 10.1111/j.1469-1809.1936.tb02137.x

[12] G. E. Forsythe, M. A. Malcolm, and C. B. Moler, *Computer Methods for Mathematical Computations*, Prentice-Hall, 1977.

[13] *Spline Interpolation Algorithm*, `http://en.wikipedia.org/wiki/Spline_interpolation`, accessed on November 4th, 2011.

[14] C. De Boor, *A practical guide to splines*, Springer 2001.

[15] *Cross-Validation and Generalized Cross-Validation*, `http://fedc.wiwi.hu-berlin.de/xplore/ebooks/html/csa/node123.html`, accessed on January 8th, 2012

[16] P. Bourke, *Bézier curves*, April 1989, updated Dec. 1996, `http://paulbourke.net/geometry/bezier/index2.html`, accessed on November 9th, 2011.

[17] R. Dias, *Nonparametric regression LOESS/LOWESS*, `http://www.ime.unicamp.br/~dias/loess.pdf`, accessed: 06/11/2011.

[18] K. Takezawa, *Introduction to nonparametric regression*, pg[185–197], John Wiley and Sons, 2006

[19] W.S. Cleveland, *A program for smoothing scatterplots by robust locally weighted regression*, The American Statistician, 1981, JSTOR

[20] *Extending Linear Regression: Weighted Least Squares, Heteroskedasticity, Local Polynomial Regression*, `www.stat.cmu.edu/~cshalizi/350/lectures/18/lecture-18.pdf`, accessed on October 15th, 2009, pg[10–12].

[21] *The National Center of Biotechnology Information*, `http://www.ncbi.nlm.nih.gov/gene/650826`, accessed on February 18th, 2012.

[22] S. Salvador and P. Chan, *Determining the number of clusters/segments in hierarchical clustering/segmentation algorithms* , Tools with Artificial Intelligence, 2004, pg[576–584], DOI: 10.1109/ICTAI.2004.50

[23] G.W. Milligan and M.C. Cooper, *An examination of procedures for determining the number of clusters in a data set*, Psychometrika, 1985, Vol. 50, Nb. 2, pg[159–179], DOI: 10.1007/BF02294245

[24] L. Kaufman and P.J. Rousseeuw, *Finding Groups in Data: An Introduction to Cluster Analysis*, 1990, DOI: 10.1002/9780470316801

[25] J. MacQueen, *Some methods for classification and analysis of multivariate observations*, Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Vol. 2, 1967, pg[281 – 298], University of California Press.

[26] J. A. Hartigan and M.A. Wong, *A K-means clustering algorithm*, Journal of the Royal Statistical Society. Series C (Applied Statistics), Vol. 28 No. 1, 1979, pg[100–108], Blackwell Publishing.

[27] S. P. Lloyd, *Least squares quantization in PCM*, IEEE Transactions on Information Theory IT, Vol. 28, 1982, pg[129–137].

[28] E. Forgy, *Cluster analysis of multivariate data: Efficiency vs. interpretability of classifications*, Biometrics 21, 768.

[29] S. Dudoit. and J. Fridlyand, *A prediction-based resampling method for estimating the number of clusters in a dataset*, Genome Biology, Vol. 3, 2002.

[30] M.Maechler, *Package 'cluster': Cluster Analysis Extended Rousseeuw et al.*, `cran.r-project.org/web/packages/cluster/cluster.pdf`, accessed on March 13th, 2012.

[31] T. Bäck, *Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms*, 1996, Oxford University Press, Inc.

[32] Y. Lu et. al, *FGKA: A Fast Genetic K-means Clustering Algorithm*, In Proc. ACM Symposium on Applied Computing, 2004, pg[622–623].

[33] K. Krishna and M. Murty, *Genetic k-means algorithm*,IEEE Trans. Syst., Man, Cybern. B, Cybern., Vol. 29, No. 3,1999, pg[433–439].

[34] A. Lucas, textitPackage 'cluster': Another Multidimensional Analysis Package, `cran.r-project.org/web/packages/amap/amap.pdf`, accessed on April 22nd, 2012.

[35] M. Barnes et al., *Experimental comparison and cross-validation of the Affymetrix and Illumina gene expression analysis platforms*, Oxford Jurnals, Nucleic Acids Research, Vol. 33, Issue 18, 2005, pg[5914–5923].

[36] The Gene Ontology Consortium, *Gene Ontology: tool for the unification of biology*, Nature Genetics, Vol. 25, 2000, pg[25–29].

[37] The Bioconductor Project, `http://www.bioconductor.org/`, accessed on 28th of March, 2012

[38] R.A. Fisher, *Statistical Methods for Research Workers*, Oliver and Boyd, 1954, ISBN 0-05-002170-2.

[39] Y. Benjamini and Y. Hochberg, *Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing*, Journal of the Royal Statistical Society. Series B, Vol. 57, No. 1, 1995, pg[289–300].