Data Integration and Clustering Analysis of Glioblastoma Multiforme for identifying expression signatures

By

Marvie DEMIT

Bachelor of Science (University of Life Science and Natural Resources in Vienna)
2019

Master Thesis

Submitted for the degree of

Diplom Ingenieur

in

Biotechnology

in the

Department of Biotechnology

of the

University of Life Science and Natural Resources in Vienna

Approved:

Priv. -Doz. Dr. Peter Sykacek

2019

Eidesstattliche Erklärung

Ich erkläre eidesstattlich, dass ich die Arbeit selbständig angefertigt, keine anderen als die angegebenen Hilfsmittel benutzt und alle aus ungedruckten Quellen, gedruckter Literatur oder aus dem Internet im Wortlaut oder im wesentlichen Inhalt übernommenen Formulierungen und Konzepte gemäß den Richtlinien wissenschaftlicher Arbeiten zitiert, durch Fußnoten gekennzeichnet bzw. mit genauer Quellenangabe kenntlich gemacht habe.

Unterschrift:

_____

Marvie DEMIT BSc.

2019

Summary

Glioblastoma Multiforme (GBM), a grade 4 glioma, develops into a rapidly growing and highly malignant tumour with very poor prognosis. The average age at diagnosis of GBM is between 40 and 60 years with GBM being more often found among male adults. The goal of this thesis is to assess the potential of machine learning based cluster analysis of expression profiles to lead to an alternative definition of GBM subtypes.

To approach the problem in an optimal manner, we propose an integrated analysis of expression data that was obtained with different measurement techniques. To obtain such data a large and well annotated collection of GBM has been downloaded from The Cancer Genome Atlas (TCGA). To warrant dependable results, alternative analysis strategies were considered at all stages of the analysis workflow. We specifically compared data integration at the level of gene expression values and subsequent clustering with clustering of individual expression data modalities and subsequent integration of cluster assignments. Clustering itself was approached by several competing methods with particular emphasis put on selecting an optimal number of clusters. The number of clusters corresponds in such analysis to the proposed number of GBM subtypes. Deciding on an optimal number of clusters is thus crucial. We put thus considerable effort in devising a collection of metrics and relied on consistency across metrics as model selection yardstick.

Careful assessments based on consistency lead to three gene expression derived GBM subtypes and corresponding patient groups which are optimal in a technical manner. To complement technical optimality with an assessment of biological relevance, we compared patient survival between the predicted groups by a Kaplan-Meier analysis

to find that the proposed GBM subtypes are significantly associated with survival. For a second analysis of biological relevance we ranked gene expression profiles by their predictive power for the established clustering and used a gene set enrichment analysis to obtain easier to comprehend biological tags to our findings. The resulting biological process and pathway terms were compared by manual literature review with established knowledge to find agreement in several biological processes which are linked with GBM development. As a conclusion of this thesis we may thus state that unsupervised machine learning has great potential to elucidate and characterise the molecular mechanisms of GBM. The proposed analysis is however far from complete as other machine learning methods are available that can replace and augment different parts of our analysis workflow. The use of data modalities like methylation signatures of copy number signals has furthermore great potential to provide additional insight.

An important result of this thesis was to establish a pipeline for integrating GBM samples, data preprocessing, clustering and post processing. Although developed specifically for the analysis of GBM subtypes, the pipeline developed in this MSc thesis may be of broader interest to communities who wish to expand their knowledge about molecular mechanisms of other types of cancer.

Zusammenfassung

Glioblastoma Multiforme (GBM), ein Tumor des Gliom-typs Grad 4, entwickelt sich zu einem schnell wachsenden und stark bösartigen Tumor. Das durchschnittliche Diagnosealter liegt zwischen 40 und 60 Jahren. Am häufigsten werden männliche Erwachsene diagnostiziert. Das Ziel dieser Arbeit ist es, umfassende Signaturen von GBM-Subtypen aus Expressionsdaten durch Implementierung von Datenintegration und Clusteranalyse zu entdecken.

Um das Problem zu lösen, wurde die Integration von Expressionsdaten aus verschiedenen Messergebnissen, die in The Cancer Genome Atlas (TCGA) öffentlich verfügbar sind, mit entsprechenden Hintergrundinformationen des Patienten abgerufen und analysiert. Um verlässliche Ergebnisse zu erzielen, werden alternative Analysestrategien in allen Abschnitten der Pipeline betrachtet - die Datenintegration auf der Ebene der Geneexpressionsdaten mit anschließendem Clustering wird mit Clustering einzelner Expressions-modalitäten und nachfolgender Integration der resultierenden Cluster-Einteilungen verglichen. Zum Clustering selbst werden mehrere Methoden herangezogen. Der sorgfältigen Modellauswahl kommt wegen der implizierten Bedutung als Anzahl von GBM subtypen besondere Bedeutung zu. Da die Optimalität entscheidend von der Modellbewertung abhängt, wurde ein erheblicher Arbeitsaufwand unternommen, um eine ganze Gruppe zuverlässiger Metriken zu bekommen, die zur Modellwauswahl verwendet werden.

Um technisch definierte Optima auf ihre biologische Relevanz hin zu untersuchen wurde auf einfach interpretierbare Terme wie biologische Prozeße und Pfade zurück gegriffen. Um Aussagen auf der Ebene solcher Terme zu machen wurden Expressionssignaturen hinsichtlich ihrer Eignung zur Vorhersage der Clusterzugehörigkeit

der GBM Patienten gereiht und anschließend mittels Gene Set Enrichment Analyse auf biologische Terme projiziert. Unter den vorhergesagten Termen konnten durch manuelle Recherche zahlreiche Prozesse identifiziert werden, die bekannte Assoziationen mit der Entwicklung von GBM haben. Als zweites Kriterium biologischer Relevanz konnten die aus den Expressionsmustern erhaltenen Patientengruppen mittels Kaplan-Meier Analyse mit signifikanten Unterschieden in deren Überlebenszeit assoziiert werden. Unsere Analysen legen nahe, dass der vorgeschlagene Einsatz maschinellem Lernens zur Gruppierung von GBM in Subtypen bzw. zum besseren Verständnis molekularer Mechanismen in GBM geeignet ist. Als Ausblick sei erwähnt, dass unsere Analyse in Zukunft um eine Reihe zusätzlicher Datenquellen ergänzt werden soll um damit das Verständnis von GBM weiter zu verbessern und in Zukunft therapeutische Ansätze zu schaffen, die momentan in der postoperativen Behandlung von GBM weitgehend wirkungslos sind.

Acknowledgements

I have been very fortunate to have Dr. Peter Sykacek as my advisor. His enthusiasm for good science of all kinds is infectious. Every quarter for almost 2 years, some of my biggest steps forward in my research, my presentation skills, and in my development as a scientist came directly from the perspective that he provided. His skills and ideas have played an important role in shaping my priorities for both statistical thinking and scientific education. All of his machine learning related courses could transformed many biotech students at the Universität for Bodenkultur in Vienna a comprehensive insight of how to think statistically as a biotechnologist. Not to forget his thoughtful approach to science, writing, programming and the big picture has provided an excellent model that I hope to follow throughout my career.

My whole family especially mentioned my mom Bebie, both of my sister Nedie and Mai, my brother in-law Christian and my two dearest nieces have consistently provided me with more support than I ever could have asked for, in every aspect of my life that allowed me to succeed. I will always be grateful to them.

Finally, I would like to thank Elizabet, for her dedication, support, and love. Whenever I've talked myself into a corner and didn't know how to proceed with my writing or research, her support has always helped me see the solutions I'd missed by giving me the directions I need in her own way. Elizabet's integrity, conscientiousness, and commitment to doing her best in all areas of life is truly inspiring, and I love her for too many reasons to list here.

Table of Contents

1

Introduction

Glioblastoma Multiforme (GBM), can be described as a grad IV glioma type tumour which develops into a rapidly growing and highly malignant tumour from a normal brain tissue with least survival period (Jemal et al., 2009). GBMs are tumours with heterogeneous characteristics, that means they expand into multiple complex genetic abnormal neoplastic cells. The average diagnosis age are between 40 and 60 years and commonly male adult patients. The risk factors and causes for the development of GBM are mostly unknown. According to current knowledge, neither environmental factors, eating habits, emotional stress, nor electromagnetic fields in the frequency range of mobile signal led to a higher risk of brain cancer(Association and others, 2016). Due to aggressiveness, insufficient treatment methods, increased number of patients and it's unknown causes and risks, it is essential to find molecular causes which impact survival of GBM patients which will provide a better understanding of the disease and may suggest an improved treatment (Association and others, 2016).

GBM subtypes

A common approach to study the genetic aberration of GBM is by gene expression analysis. The technique can be used for individual transcripts as well as for the entire transcriptome and allows for quantitative statements about the activity of genes (DeRisi et al., 1997). Gene expression levels can be measured at the level of RNA or proteins. The possibility of gene expression analysis has led to the identification of prognostic profiles, some of which have been validated and are in clinical use. In addition, clinical factors such as tumour size, etc. are important information about tumour prognosis (DeRisi et al., 1997).

After the first treatment of GBM by surgery, prognostic and predictive factors are assessed to identify the best suitable treatment. By analysing gene expression profiles, molecular subtypes of GBM were identified that differ significantly in their clinical course and response. Gene expression profiles aim to provide prognostic information beyond conventional clinico-pathological risk taking to influence the therapy decision if necessary (Wallner et al., 1989). (Verify the validity of citations of Wallner)

Unsupervised learning methods have been applied to gene expression data to identify relationships among genes (Murat et al., 2008). Unsupervised learning methods can however also discover commonalities among patients which may identify subtypes of diseases in general (Murat et al., 2008) and GBM in particular. By associating gene expression signatures with such patient clusterings, we may provide a deeper insight into the molecular characteristics of the disease (Murat et al., 2008).

Cluster analysis is a very important technique in statistics and machine learning. Therefore, a wide range of different methods with individual strengths and weaknesses can be found. Due to the well known ("no free lunch theorem") from D. Wolpert and W. Macready (Wolpert and Macready, 1997), it is impossible to favour a particular method a-priori. Clustering should thus be approached by several competing methods. Particular emphasis should be given to careful model selection which decides on the optimal number of clusters. As cluster optimality depends on the metric, we make an exceptional effort on implementing and applying different validation metrics which are subsequently used as model selection criteria (Van't Veer et al., 2002).

Established GBM subtypes and challenges

Different researchers proposed different classifications of GBM, which can be distinguished by their genomic features, survival period, patient and treatment responses. Phillips et al. categorised GBM by 3 groups: mesenchymal (49%), Proneural (31%) and proliferative (20%). While Mesenchymal and proliferative subtypes has equally short survival period, Proneural subtype was associated with patients with longer survival period (Phillips et al., 2006). The study by Verhaak et al. used a large patient cohort of GBM samples which can now be obtained from TCGA GDC. Their analysis identified four distinct groups of GBM: Mesenchymal, Proneural, Classical and Neural subtypes. A characterisation of genetic differences and clinical outcome is provided in figure 1.1. Their initial research was performed with 206 GBM patient sample data containing 601 genes from 91 patients with mutations in TP53. The classical subtype is classified with no TP53 mutations but high EGFR amplification around 97%. The mesenchymal subtype has mutations in NF1 (38%), TP53 (3%) and PTEN (87%). In addition Verhaak et. al. (Verhaak et al., 2010) report low expression for NF1 and high expression of TNF and NF-KB.

A later study of Purkait et al. used 114 GBM patients which were diagnosed from 2006 until 2012 in the Neuropathology Laboratory of the All India Institute of Medical Science. The authors find not correlation of the current classification by Verhaak et. al. with the clinical outcome of their study (Purkait et al., 2016). Purkait et. al. raised some concerns about the Verhaak et. al. classification by finding that protein expression measured with immunohistochemistry shows no correlation with gene amplification of EGFR, PDGFR or TP53 (Purkait et al., 2016).

This discrepancy suggests that a careful analysis of the TCGA GBM data should be carried out with emphasis on finding interesting patterns which allow separating the patient cohort into subgroups which share distinct molecular features. Since we do not have a priori information about patient groups, this analysis should be approached with methods which belong to the category of unsupervised machine learning methods.

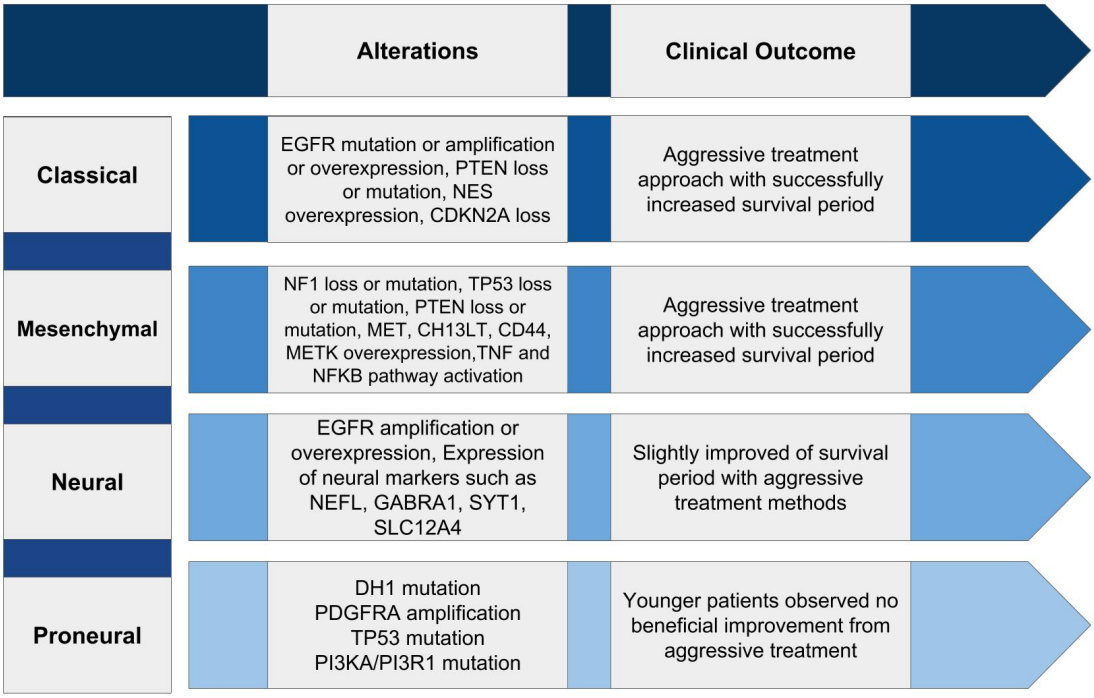

Figure 1.1: GBM Subtypes by (Verhaak et al., 2010). The figure summarises genetic characterisations and observed gene expression abnormalities which are characteristic for the four GBM subtypes as identified by (Verhaak et al., 2010). In addition, the figure summarises typical clinical phenotypes of patients which suffer from the different GBM subtypes. A common aspect of all GBM subtypes is very poor prognosis of patients and lack of promising treatment. This suggests that further analysis into GBM which could improve its characterisation is timely needed.

Research objective of this work

This master thesis proposes a careful clustering of GBM patients with the goal to evaluate the potential of unsupervised machine learning to overcome known shortcomings of current GBM subtypes. We hypothesise that this objective can be achieved if we apply carefully tuned unsupervised methods on a large set of GBM gene expression profiles. To obtain the required data, we downloaded all available GBM cases from TCGA GDC where gene expression data was available. This gave rise to three different measurement platforms. To evaluate our hypothesis, we need therefor a carefully designed workflow. To remove platform biases and other detrimental components

from the gene expression data, we used methods from the bioconductor framework for normalisation and quality control (Gentleman et al., 2004). To allow robust conclusions, we combine information from different data modalities by two alternative strategies:

- We may combine information from different data sources by careful alignment of samples and variables. Data may be integrated at gene expression level, for example by averaging. We propose this approach for its similarity to factor analysis approach which was applied in (Verhaak et al., 2010) for combining information. A disadvantage of this choice is loss of information by having to restrict analysis to samples and genes where all modalities are available. An additional complication arrises from platform specific biases which render inter platform normalisation challenging.
- We propose therefore a second approach which combines data at meta level. For deciding for gene subsets which convey information angoput GBM, we combine p-values using Fishers meta analysis (Fisher, 2006). Once we decided for a gene set all further processing and clustering can be done separately for every data modality. This has the advantage that we avoid loss of information as we may use all cases. Once all individual clusterings are available, the overlapping cases are used for resolving the identification problem between the different clusterings (Stephens, 2000). After resolving we may combine the probabilistic cluster assignments quantitatively and arrive at consensus clusters.

It is evident that reliable clustering is an important aspect to both approaches with the optimal number of cluster centres corresponding to the proposed number of GBM subtypes. Warranting a reliable cluster number determination is thus the by far most important aspect of our analysis pipeline. To achieve this goal we spent considerable effort on surveying theory, metric validation and application. To validate our hypothesis we used two strategies for assessing the obtained clusterings:

- The cluster indicators of samples were used as factors in a linear model to obtain characteristic expression signatures and gene rankings. By using a GSEA type method, the rank lists were tagged with easy to comprehend terms from the Gene Ontology and biological pathway databases. For verifying the biological implications suggested by this analysis, we compared our findings by careful

review to find good agreement with biological implication that are linked to GBM in published literature.

- The cluster indicators were also used to separate clinical parameters into groups. We investigated in particular the connection between our predicted GBM clusters and patient survival and discovered a distinct separation among short to medium survival periods. Significance was however dependent on the applied clustering method.

Our findings allow the conclusion that a careful analysis of expression data by clustering and quantitative integration of evidence provides molecular distinct subgroups of samples which are linked with meaningful biological processes and different clinical prognosis. There are however also several aspects wich can be improved TCGA has several other modalities like genotype, methylation state or copy number variation which will carry information about GBM subtypes. These data sources should be considered for integration.

By subjecting the gene expression signature which is significantly linked to our proposed clustering of GBM to a GSEA analysis we find biological terms like "control loss of cell cycle", "regulation of G1-S transition", "fibroblast growth factor receptor signalling pathway" and "signal transduction by p53 class mediator" among the top ranked GO terms. A careful literature analysis reveals these terms linked with development of GBM (Nakada et al., 2011). Albeit observing such confirmatory evidence of our clustering, there is one aspect which deserves a follow-up: The gene list we obtain when using the cluster indicators as rank effect finds for all measurement platforms strong differential expression for a large fraction of genes. This implies a string variability of the molecular signatures in dependence of GBM subtype. It is clearly necessary to investigate this result further to decide whether this is interesting biology or a side effect from an unwanted batch effect which is consistently present in all gene expression platforms and should thus be removed.

2

Glioblastoma Multiforme

Gliomas: Occurrence and Classifications

Most common gliomas are high-grade gliomas and the occurrence of primary brain tumours are increasing. Despite that gliomas are the minor widespread tumour types, they are the main cause relating to cancer deaths in Europe under 40. Brain cancer occurrence cannot be prevented by any kind of habits adjustment and no compelling development in survival rates for almost 30 years has been classified yet(Weller et al., 2014).

The name glioma comes after the cells from which glioma occurs and develops and they make up about ~50% of the entire primary brain cancers and is composed of wide distribution of neuroectodermal tumours, containing ependymomas, oligodendrogliomas and astrocytomas. Astrocytomas that occurs from astrocytes which are the most frequent types of the brain cells and consist the most glioma groups, are above 75 percent(Weller et al., 2014). Grade I astrocytomas or also known as pilocytic astrocytomas are mostly curable by surgical extraction procedure. Grade II astrocytomas with 6 to 8 years of overall survival, have slow growth rate character and low proliferation as a result (Crocetti et al., 2012). Grade III astrocytomas, also

known as anaplastic astrocytomas, depending on the last grade progression, have a median overall survival of ~3 years. Grade IV astrocytomas which are more classified as glioblastoma multiforme (GBM), in comparison to the lower grade gliomas, differ in prominent vascular proliferation, pseudopalisading necrosis and high growth rate. The majority of lower grade II - III astrocytomas rapidly arise into a primary GBM or de novo GBM without a less malignant precursor lesion (Ohgaki and Kleihues, 2012). As the result the amount of fluorescence intensity from every probe on the plate of the expression is measure via fluorescence measurement.

Development of GBM

The large part of Glioblastoma Multiforme GBM tumours are being observed in the supratentorial cell areas, in the frontal lobes, although they can be also found in cortical parts, the brainstem, spinal cord and the cerebellum (Adamson et al., 2009). GBM tumours usually develop in the central nervous system (CNS) and in spite of GBM tumours as being highly aggressive, they exceptionally contribute outside the CNS. GBM tumours contain genetically and phenotypically heterogeneous groups of tumours that occur in average of about 4 cases from 100,000 people across Europe. The average age is around 50 years and above and mostly affected patients are male (3:1) (Crocetti et al., 2012).

Because of the heterogeneous character of GBM, the histopathology of the tumour is extremely complex. Morphologically, GBM can be visually identified as grey coloured tumour cells with yellow stained necrotic tissue by cause of myelin disruption and variety of hemorrhagic areas. One of the features that has been recognised as a prognostic feature for differing the GBMs between the lower grade gliomas are including pseudopalisading necrosis, a composition that is particularly unique to GBM and microvascular hyperplasia that is linked with the pseudopalisades development. Pseudopalisading necrosis, where hyper-cellular areas, depart from the hypoxic necrotic regions, where the cells over-expressed hypoxia-inducible factor (HIF-1), introducing cell responses to low oxygen concentration and inducing pro-angiogenic factors for instance vascular endothelial growth factor (VEGF) (Rong et al., 2006).

GBMs are generally classified into primary and secondary GBM that influence a vari-

ety of age groups, develop at varying rates and affecting multiple genetic conversions on clinical basis(Masui et al., 2012). Although primary and secondary GBMs share common clinical progress, they differ in molecular pathways such as the Ras/MAPK cascade, PI3K and p53 pathways. While older patients are mostly affected with primary GBMs, secondary GBM are found in most younger patients below the age of 45 years. 90% of GBM arise de novo by multistep tumorigenesis from normal glial cells (primary GBM), their presence occur rapidly and are usually easily detectable. Additionally, they represent as the most aggressive form of GBM. About 55 percent of GBMs are described by epidermal growth factor receptor (EGFR) over-expression due to genetic (Henriksen et al., 2014). The most part of GBMs grow immediately without any clinical indication of precursor lesion. The development of lower grade gliomas into secondary GBM (10% of GBM) requires genetic transformations which is described in the figure 2.3. The most common genetic modification of GBM, which cover up 60% to 80% of most cases is the loss of heterozygosity (LOH) on chromosome 10q. The majority of primary GBM indicate loss of the entire chromosome, when in fact the secondary GBMs show partly loss of 10q (Brat and Van Meir, 2004). MGMT promoter methylation has been much more observed in secondary GBMs around 75% than in primary GBMs which is about 36%. On chromosome 9, p16INK4a and p14ARF suppressor genes are encoded within the CDKN2A locus.

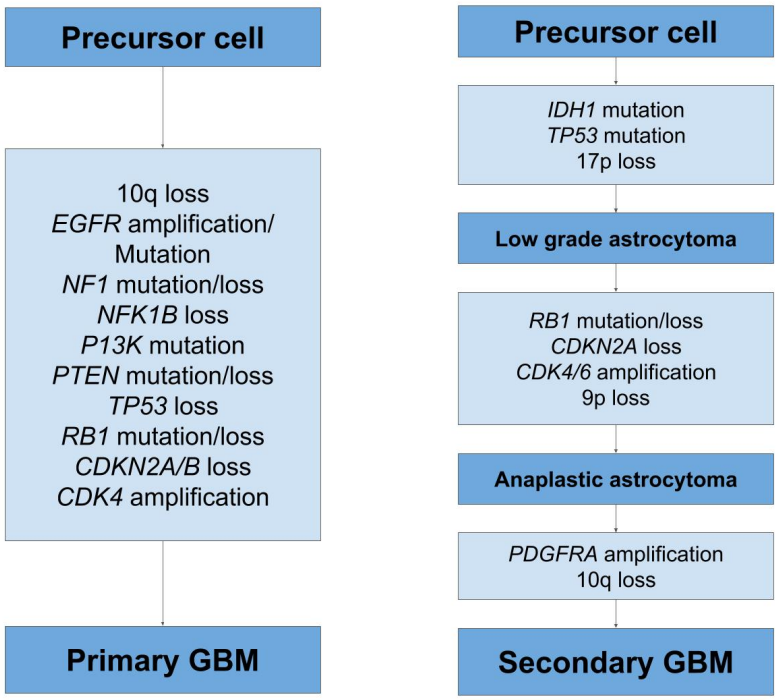

Figure 2.1: Differences of the development between primary and secondary GBM. primary GBMs arise de novo, whereas secondary GBMs arise from lower grade lesions.

Generally, the diagnosis of GBM is based on histopathological methods. Nowadays, due to GBMs heterogeneity, the result of improper sampling, molecular markers are turning into more reliable methods preferred in routine diagnostics. Our current understanding of molecular events, as with most cancers, suggests that unlikely all patients share a common genetic trait. Isocitrate dehydrogenase-1 (IDH1) mutations have been identified in most low grade tumours and considered as an early event in gliomagenesis (Nikiforova and Hamilton, 2011). Secondary GBMs have high rates of IDH1 mutations contributing to a better prognosis (Dunn et al., 2013)

While IDH1 mutations in the primary GBM are rare, EGFR amplifications (40-60%) and PTEN (15-40%) are more common (Kanu et al., 2009). TP53 mutations are more common in secondary GBMs ($>$ 60%) than primary GBMs ($\sim$ 10%) and are the earliest detectable genetic mutation present in 60% of low-grade progenitor astrocytomas. However, the TP53 and RB1 signalling pathways are often altered in the primary GBM (Zhu and Parada, 2002). Therefore, excelling approach of characterisations between the types of astrocytoma tumours should be defined properly in

order to achieve better treatments within the next years (Kunkle et al., 2013).

Cell biology of GBM

In the last two decades the biological structures and functions of GBM have been thoroughly researched. GBM cells are subjected to different kinds of cell malfunction that gain advantage resisting different kinds of anti-GBM treatments. Six cellular characterisations of GBM are shown below but it should be recognised that these characterisations do not occur in isolation (Nakada et al., 2011).

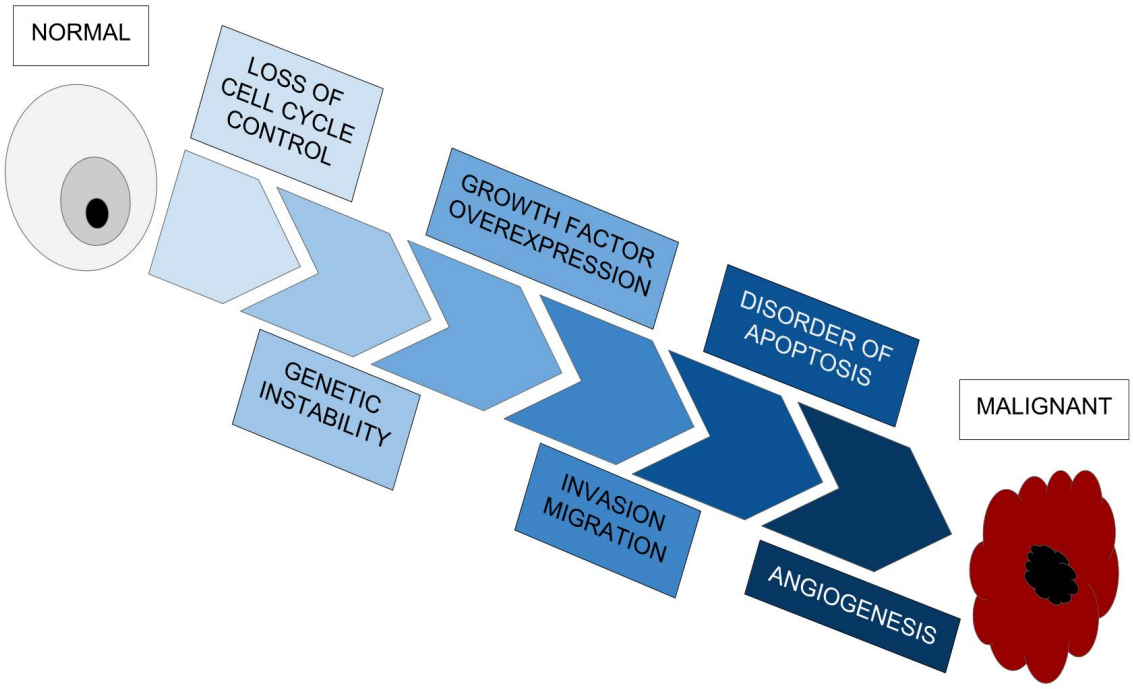

Figure 2.2: Gliomagenesis. Progressive accumulation of tumour transformation in GBM is subjected in multiple intracellular events. Starting from a normal healthy cell, it undergoes different events such as loss of cell cycle control due to lack of regulators that coordinates the cell division. Genomic instability is one of the main events in most cancer due to mutations in the DNA repair genes, which promotes cancer development from healthy cells.

The first occurrence is loss of cell cycle control. The normal cell cycle is immensely strict regulated. However, these regulations are being inhibited by the glioma tu-

mours, which cause genetic defects in growth regulatory factors, allowing them limitless proliferation. These genetic defects are observed in malignant rather than lower grade glioma cells. The G1 - S phase transition has been remarkably noticed, since changes occur in one or more components of p16INK4a/cyclin-dependent kinase (CDK)-4/RB (retinoblastoma) 1 pathway, regulating G1-S phase transition cell cycle checkpoint in various anaplastic astrocytomas. RB1 undergoes phosphorylation by CDK/cyclin D1 complex, which activate genes in G1 - S phase transition, releasing E2F transcription factor (Nakada et al., 2011).

Over-expression of cellular growth factors and their receptors, the second event of the cell cycle pathway, are the main cause of GBM development. Diverse growth factors are over-expressed in GBM and transducing cell proliferation and turn healthy cells into neoplastic tumours such as epidermal growth factor receptor (EGFR), platelet-derived growth factor (PDFG), basic fibroblast growth factor (bFGF, FGF-2), transforming growth factor (TGF)-α, and insulin-like growth factor (IGF-1). Among the growth factors that are over-expressed in GBM, EGFR and PDGF are the main over-expressed proteins (Nakada et al., 2011).

Angiogenesis is also an important part of causing and sustaining GBM. The rapidly growing tumours in GBM development are surrounded by angiogenic alterations that occur as ring-like contrast enhancements. These alterations are visible at the Magnetic Resonance Imaging (MRI) scan. Specially in GBMs angiogenic molecules are present in malignant gliomas. As a result of microvascular proliferation, malignant gliomas are vascular tumours (Nakada et al., 2011).

Also one of the event that causing and sustaining GBMs are invasion and migration events that are influenced by extracellular matrix molecules (ECM) and cell surface receptors and these influence the GBM to diffuse and infiltrate of the surrounding neural net. The cytoskeletal proteins are included; signalling molecules that resolve the communication between the microenvironment and the cytoskeleton (Nakada et al., 2007).

The next event that should be mentioned and is also a key feature to the cell cycle and that is the abnormality of apoptosis which is characterised as a programmed cell death by non-inflammatory cellular condensation. Glioma cells develop means for increased proliferation and to abrogating apoptosis. The apoptotic response

in normal glial disturb by p53 mutations that usually follow growth factor over-expression in low-grade gliomas, leading to progressive development (Nakada et al., 2007).

A vital role and the last event is the genetic instability of GBM development. A crucial feature of low-grade glioma is the rapid progress to high-grade lesions and such malignant progression is correlated to the malignant clones development. More Malignant clones are selected occurring further genomic damages as a result of genomic instability. Mutations in p53, also called as "guardian of the genome", may cause tumour progression through genomic instability. Patients have an increased of developing malignant gliomas with syndromes of genomic instability (Nakada et al., 2007).

Causes and Symptoms of GBM

The primary cause of GBM is still unknown. Most brain tumours are genetically inheritable, though genetically heritable diseases such as Tuberous Sclerosis, Neurofibromatosis, Li-Fraumeni and Von Hippel-Lindau occasionally trigger the cancer tumour, which only a small quantity of cases have been recorded of being activated by the genetically heritable diseases mentioned. Recently, researchers speculate that abnormalities (genetically and immunogenicity), environmental factors such as UV-lights exposures, ionising radiations and stress along with other factors that cause deformities in genes of various chromosomes are responsible for triggering the tumour development. However, no significant confirmations has been delivered a direct correlation between the factors and the development of GBM tumours within those cases. Investigators are approaching ongoing fundamental research to study more about the underlying factors of causing GBM (Association and others, 2016).

Abnormal changes of cell structure or loss of tumour suppressor gene are one of the causes for GBM development secondary to the oncogenes (tumour suppressor genes regulates cell division) which control cell growth. The main cause to this specific development is not clarified. Nonetheless, latest studies proposed that abnormalities of DNA (deoxyribonucleic acid), which carries the gene information, are the fundamentals of malignant cell transformation (Nakada et al., 2011).

The typically inversion to a more primitive form of tumour (loss of differentiation or anaplasia) in cells that developed malignancies which is the result of incapability performing their respective functions within the tissue (Nakada et al., 2011). Once cells developed malignancies, they pass these abnormalities to their "daughter" cells with a rate of a rapidly and uncontrolled division, which the natural immune defences of the body are incapable to compete. Ultimately, the formation of a mass known as tumour or neoplasm is the result due to such uncontrolled proliferation and abnormalities of the cells. Thus, cells proliferate heterogeneously, which means that the cells within the tumour do not share identical genetic properties(Nakada et al., 2011).

The location of the neoplasm, the growth size and the growing rate are the dependencies of the symptoms displayed from the patients. There are cases when symptoms occurs directly after tumour development, however in more of the cases, symptoms only occurs when the tumour has reached a definite size (Nakada et al., 2011).

Based on patient records, general symptoms of GBM are headaches with different stages of intensities, usually occurs after sleeping, early mornings that leads to nausea and vomiting issues and in later periods to hemiparesis (a one side paralysis of the body), loss of motoric skills and affective sensation. The cognitive perception is also negatively affected, adverse concentration and mental development, loss of visual capability and aphasia (language dysfunction) (Association and others, 2016).

Prognosis and Treatment of GBM

A suitable diagnosis has to be subjected on patients being detected with this kind of tumour before it can be treated. The initiation of diagnosis is to execute a neurological observation on the patient and afterwards performing a Magnetic Resonance Imaging (MRI), Computed Tomography (CT) or Magnetic Resonance Spectroscopy (MRS) scan. These methods are essential for tumour location, size, tumour type, mineral and chemical measurements in result to the malignancy results of the patients (Association and others, 2016).

GBM is a genetically heterogeneous neoplastic tumour with complex structures as the results its existence of sub-clones within the tumour cell population. The existence

of sub-clones and their heterogeneity has made GBM resistant to the introduced treatment methods. The conventional GBM treatment method has been unchanged for years. A surgical intervention is performed on the patient to extract the tumour, secondly a radiation therapy and subsequently the chemotherapy is executed. In most of the cases, the average survival of those GBM diseased patients are about nine to ten months even after all visible MRI scanned tumours have been surgically extracted and being treated with radiation and chemotherapy. This is due to the diffusive topography that makes the tumour location inconsistent that leads to unsuitable resection of the tumour (E.C. Holland 2000). It has not been able to fully undergo full resection with any adverse neurological and functional side effects such as motoric disorders, that could impact the quality of living (Von Neubeck et al., 2015).

In spite of the aggressiveness of the disease and despite to the technological development obtained in surgery, radio- and chemotherapy, the survival periods of treated patients has been marginally improved. Even though with these intensive treatment applications to the GBM, resistance has been observed despite to the intensive multimodal therapy methods and the survival period just slightly increased with just couple of months. The treatment proposal was introduced by the European Organisation for Research and Treatment of Cancer (EORTC) and National Cancer Institute of Canada Clinical Trials Group (NCIC). This approach implicates surgical procedure for extracting all the tumours followed by fractionated radiotherapy beside of concomitant and adjuvant treatment of temozolomide (TMZ) a cytostatic agent. The median has been increased due to this applied method to the patients with 2 years survival up to 14.6 months and 26.5% compared to patients only treated with radiotherapy, which is only 10.4% and 12.1 months (Von Neubeck et al., 2015). # Glioblastoma Multiforme

Gliomas: Occurrence and Classifications

Most common gliomas are high-grade gliomas and the occurrence of primary brain tumours are increasing. Despite that gliomas are the minor widespread tumour types, they are the main cause relating to cancer deaths in Europe under 40. Brain cancer occurrence cannot be prevented by any kind of habits adjustment and no compelling development in survival rates for almost 30 years has been classified yet(Weller et

al., 2014).

The name glioma comes after the cells from which glioma occurs and develops and they make up about ~50% of the entire primary brain cancers and is composed of wide distribution of neuroectodermal tumours, containing ependymomas, oligodendrogliomas and astrocytomas. Astrocytomas that occurs from astrocytes which are the most frequent types of the brain cells and consist the most glioma groups, are above 75 percent(Weller et al., 2014). Grade I astrocytomas or also known as pilocytic astrocytomas are mostly curable by surgical extraction procedure. Grade II astrocytomas with 6 to 8 years of overall survival, have slow growth rate character and low proliferation as a result (Crocetti et al., 2012). Grade III astrocytomas, also known as anaplastic astrocytomas, depending on the last grade progression, have a median overall survival of ~3 years. Grade IV astrocytomas which are more classified as glioblastoma multiforme (GBM), in comparison to the lower grade gliomas, differ in prominent vascular proliferation, pseudopalisading necrosis and high growth rate. The majority of lower grade II - III astrocytomas rapidly arise into a primary GBM or de novo GBM without a less malignant precursor lesion (Ohgaki and Kleihues, 2012). As the result the amount of fluorescence intensity from every probe on the plate of the expression is measure via fluorescence measurement.

Development of GBM

The large part of Glioblastoma Multiforme GBM tumours are being observed in the supratentorial cell areas, in the frontal lobes, although they can be also found in cortical parts, the brainstem, spinal cord and the cerebellum (Adamson et al., 2009). GBM tumours usually develop in the central nervous system (CNS) and in spite of GBM tumours as being highly aggressive, they exceptionally contribute outside the CNS. GBM tumours contain genetically and phenotypically heterogeneous groups of tumours that occur in average of about 4 cases from 100,000 people across Europe. The average age is around 50 years and above and mostly affected patients are male (3:1) (Crocetti et al., 2012).

Because of the heterogeneous character of GBM, the histopathology of the tumour is extremely complex. Morphologically, GBM can be visually identified as grey

coloured tumour cells with yellow stained necrotic tissue by cause of myelin disruption and variety of hemorrhagic areas. One of the features that has been recognised as a prognostic feature for differing the GBMs between the lower grade gliomas are including pseudopalisading necrosis, a composition that is particularly unique to GBM and microvascular hyperplasia that is linked with the pseudopalisades development. Pseudopalisading necrosis, where hyper-cellular areas, depart from the hypoxic necrotic regions, where the cells over-expressed hypoxia-inducible factor (HIF-1), introducing cell responses to low oxygen concentration and inducing pro-angiogenic factors for instance vascular endothelial growth factor (VEGF) (Rong et al., 2006).

GBMs are generally classified into primary and secondary GBM that influence a variety of age groups, develop at varying rates and affecting multiple genetic conversions on clinical basis(Masui et al., 2012). Although primary and secondary GBMs share common clinical progress, they differ in molecular pathways such as the Ras/MAPK cascade, PI3K and p53 pathways. While older patients are mostly affected with primary GBMs, secondary GBM are found in most younger patients below the age of 45 years. 90% of GBM arise de novo by multistep tumorigenesis from normal glial cells (primary GBM), their presence occur rapidly and are usually easily detectable. Additionally, they represent as the most aggressive form of GBM. About 55 percent of GBMs are described by epidermal growth factor receptor (EGFR) over-expression due to genetic (Henriksen et al., 2014). The most part of GBMs grow immediately without any clinical indication of precursor lesion. The development of lower grade gliomas into secondary GBM (10% of GBM) requires genetic transformations which is described in the figure 2.3. The most common genetic modification of GBM, which cover up 60% to 80% of most cases is the loss of heterozygosity (LOH) on chromosome 10q. The majority of primary GBM indicate loss of the entire chromosome, when in fact the secondary GBMs show partly loss of 10q (Brat and Van Meir, 2004). MGMT promoter methylation has been much more observed in secondary GBMs around 75% than in primary GBMs which is about 36%. On chromosome 9, p16INK4a and p14ARF suppressor genes are encoded within the CDKN2A locus.

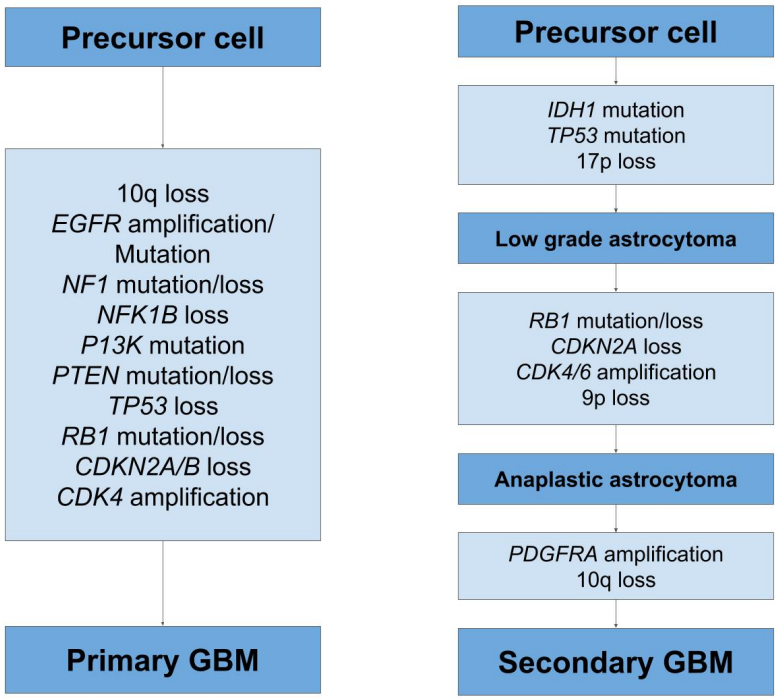

Figure 2.3: Differences of the development between primary and secondary GBM. primary GBMs arise de novo, whereas secondary GBMs arise from lower grade lesions.

Generally, the diagnosis of GBM is based on histopathological methods. Nowadays, due to GBMs heterogeneity, the result of improper sampling, molecular markers are turning into more reliable methods preferred in routine diagnostics. Our current understanding of molecular events, as with most cancers, suggests that unlikely all patients share a common genetic trait. Isocitrate dehydrogenase-1 (IDH1) mutations have been identified in most low grade tumours and considered as an early event in gliomagenesis (Nikiforova and Hamilton, 2011). Secondary GBMs have high rates of IDH1 mutations contributing to a better prognosis (Dunn et al., 2013)

While IDH1 mutations in the primary GBM are rare, EGFR amplifications (40-60%) and PTEN (15-40%) are more common (Kanu et al., 2009). TP53 mutations are more common in secondary GBMs ($>$ 60%) than primary GBMs ($\sim$ 10%) and are the earliest detectable genetic mutation present in 60% of low-grade progenitor astrocytomas. However, the TP53 and RB1 signalling pathways are often altered in the primary GBM (Zhu and Parada, 2002). Therefore, excelling approach of characterisations between the types of astrocytoma tumours should be defined properly in

order to achieve better treatments within the next years (Kunkle et al., 2013).

Cell biology of GBM

In the last two decades the biological structures and functions of GBM have been thoroughly researched. GBM cells are subjected to different kinds of cell malfunction that gain advantage resisting different kinds of anti-GBM treatments. Six cellular characterisations of GBM are shown below but it should be recognised that these characterisations do not occur in isolation (Nakada et al., 2011).

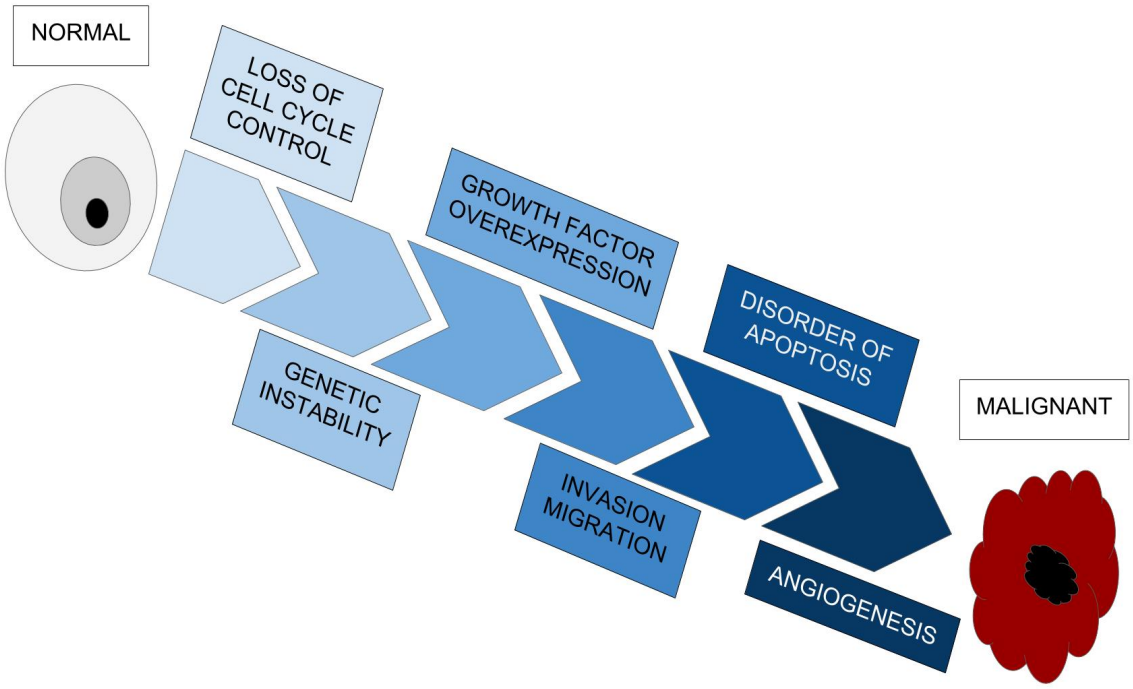

Figure 2.4: Gliomagenesis. Progressive accumulation of tumour transformation in GBM is subjected in multiple intracellular events. Starting from a normal healthy cell, it undergoes different events such as loss of cell cycle control due to lack of regulators that coordinates the cell division. Genomic instability is one of the main events in most cancer due to mutations in the DNA repair genes, which promotes cancer development from healthy cells.

The first occurrence is loss of cell cycle control. The normal cell cycle is immensely strict regulated. However, these regulations are being inhibited by the glioma tu-

mours, which cause genetic defects in growth regulatory factors, allowing them limitless proliferation. These genetic defects are observed in malignant rather than lower grade glioma cells. The G1 - S phase transition has been remarkably noticed, since changes occur in one or more components of p16INK4a/cyclin-dependent kinase (CDK)-4/RB (retinoblastoma) 1 pathway, regulating G1-S phase transition cell cycle checkpoint in various anaplastic astrocytomas. RB1 undergoes phosphorylation by CDK/cyclin D1 complex, which activate genes in G1 - S phase transition, releasing E2F transcription factor (Nakada et al., 2011).

Over-expression of cellular growth factors and their receptors, the second event of the cell cycle pathway, are the main cause of GBM development. Diverse growth factors are over-expressed in GBM and transducing cell proliferation and turn healthy cells into neoplastic tumours such as epidermal growth factor receptor (EGFR), platelet-derived growth factor (PDFG), basic fibroblast growth factor (bFGF, FGF-2), transforming growth factor (TGF)-α, and insulin-like growth factor (IGF-1). Among the growth factors that are over-expressed in GBM, EGFR and PDGF are the main over-expressed proteins (Nakada et al., 2011).

Angiogenesis is also an important part of causing and sustaining GBM. The rapidly growing tumours in GBM development are surrounded by angiogenic alterations that occur as ring-like contrast enhancements. These alterations are visible at the Magnetic Resonance Imaging (MRI) scan. Specially in GBMs angiogenic molecules are present in malignant gliomas. As a result of microvascular proliferation, malignant gliomas are vascular tumours (Nakada et al., 2011).

Also one of the event that causing and sustaining GBMs are invasion and migration events that are influenced by extracellular matrix molecules (ECM) and cell surface receptors and these influence the GBM to diffuse and infiltrate of the surrounding neural net. The cytoskeletal proteins are included; signalling molecules that resolve the communication between the microenvironment and the cytoskeleton (Nakada et al., 2007).

The next event that should be mentioned and is also a key feature to the cell cycle and that is the abnormality of apoptosis which is characterised as a programmed cell death by non-inflammatory cellular condensation. Glioma cells develop means for increased proliferation and to abrogating apoptosis. The apoptotic response

in normal glial disturb by p53 mutations that usually follow growth factor over-expression in low-grade gliomas, leading to progressive development (Nakada et al., 2007).

A vital role and the last event is the genetic instability of GBM development. A crucial feature of low-grade glioma is the rapid progress to high-grade lesions and such malignant progression is correlated to the malignant clones development. More Malignant clones are selected occurring further genomic damages as a result of genomic instability. Mutations in p53, also called as "guardian of the genome", may cause tumour progression through genomic instability. Patients have an increased of developing malignant gliomas with syndromes of genomic instability (Nakada et al., 2007).

Causes and Symptoms of GBM

The primary cause of GBM is still unknown. Most brain tumours are genetically inheritable, though genetically heritable diseases such as Tuberous Sclerosis, Neurofibromatosis, Li-Fraumeni and Von Hippel-Lindau occasionally trigger the cancer tumour, which only a small quantity of cases have been recorded of being activated by the genetically heritable diseases mentioned. Recently, researchers speculate that abnormalities (genetically and immunogenicity), environmental factors such as UV-lights exposures, ionising radiations and stress along with other factors that cause deformities in genes of various chromosomes are responsible for triggering the tumour development. However, no significant confirmations has been delivered a direct correlation between the factors and the development of GBM tumours within those cases. Investigators are approaching ongoing fundamental research to study more about the underlying factors of causing GBM (Association and others, 2016).

Abnormal changes of cell structure or loss of tumour suppressor gene are one of the causes for GBM development secondary to the oncogenes (tumour suppressor genes regulates cell division) which control cell growth. The main cause to this specific development is not clarified. Nonetheless, latest studies proposed that abnormalities of DNA (deoxyribonucleic acid), which carries the gene information, are the fundamentals of malignant cell transformation (Nakada et al., 2011).

The typically inversion to a more primitive form of tumour (loss of differentiation or anaplasia) in cells that developed malignancies which is the result of incapability performing their respective functions within the tissue (Nakada et al., 2011). Once cells developed malignancies, they pass these abnormalities to their "daughter" cells with a rate of a rapidly and uncontrolled division, which the natural immune defences of the body are incapable to compete. Ultimately, the formation of a mass known as tumour or neoplasm is the result due to such uncontrolled proliferation and abnormalities of the cells. Thus, cells proliferate heterogeneously, which means that the cells within the tumour do not share identical genetic properties(Nakada et al., 2011).

The location of the neoplasm, the growth size and the growing rate are the dependencies of the symptoms displayed from the patients. There are cases when symptoms occurs directly after tumour development, however in more of the cases, symptoms only occurs when the tumour has reached a definite size (Nakada et al., 2011).

Based on patient records, general symptoms of GBM are headaches with different stages of intensities, usually occurs after sleeping, early mornings that leads to nausea and vomiting issues and in later periods to hemiparesis (a one side paralysis of the body), loss of motoric skills and affective sensation. The cognitive perception is also negatively affected, adverse concentration and mental development, loss of visual capability and aphasia (language dysfunction) (Association and others, 2016).

Prognosis and Treatment of GBM

A suitable diagnosis has to be subjected on patients being detected with this kind of tumour before it can be treated. The initiation of diagnosis is to execute a neurological observation on the patient and afterwards performing a Magnetic Resonance Imaging (MRI), Computed Tomography (CT) or Magnetic Resonance Spectroscopy (MRS) scan. These methods are essential for tumour location, size, tumour type, mineral and chemical measurements in result to the malignancy results of the patients (Association and others, 2016).

GBM is a genetically heterogeneous neoplastic tumour with complex structures as the results its existence of sub-clones within the tumour cell population. The existence

of sub-clones and their heterogeneity has made GBM resistant to the introduced treatment methods. The conventional GBM treatment method has been unchanged for years. A surgical intervention is performed on the patient to extract the tumour, secondly a radiation therapy and subsequently the chemotherapy is executed. In most of the cases, the average survival of those GBM diseased patients are about nine to ten months even after all visible MRI scanned tumours have been surgically extracted and being treated with radiation and chemotherapy. This is due to the diffusive topography that makes the tumour location inconsistent that leads to unsuitable resection of the tumour (E.C. Holland 2000). It has not been able to fully undergo full resection with any adverse neurological and functional side effects such as motoric disorders, that could impact the quality of living (Von Neubeck et al., 2015).

In spite of the aggressiveness of the disease and despite to the technological development obtained in surgery, radio- and chemotherapy, the survival periods of treated patients has been marginally improved. Even though with these intensive treatment applications to the GBM, resistance has been observed despite to the intensive multimodal therapy methods and the survival period just slightly increased with just couple of months. The treatment proposal was introduced by the European Organisation for Research and Treatment of Cancer (EORTC) and National Cancer Institute of Canada Clinical Trials Group (NCIC). This approach implicates surgical procedure for extracting all the tumours followed by fractionated radiotherapy beside of concomitant and adjuvant treatment of temozolomide (TMZ) a cytostatic agent. The median has been increased due to this applied method to the patients with 2 years survival up to 14.6 months and 26.5% compared to patients only treated with radiotherapy, which is only 10.4% and 12.1 months (Von Neubeck et al., 2015). # Materials and Methods

Gene expression analysis

Microarray technology and RNA-Seq are important tools in global gene expression analysis. Hereby the expression of several hundred genes can be investigated simultaneously. In individual tumours, important cell biology relationships in tumour cells can be detected by gene expression analysis (Miller and Tang, 2009).

The most important role of these technologies in the future lies in the classification and prognostic assessment of diseases and gene expression. For individual disease subtypes or risk groups, specific gene expression profiles can be established (Pagliarulo et al., 2002). With this information, it is then possible - in addition to the previously available morphological and molecular genetic criteria - to improve the classification and prognosis estimation of tumour diseases. Another area of application of global gene expression analysis is the "prediction" of drug efficacy. Studies suggest that it will be possible to generate gene expression profiles of body cells that are specific to the responses of the cells to drug therapy. In the future, this may be important for the individual therapy of tumour diseases (Van't Veer et al., 2002).

Most accurate characterisation for classification, prognosis and treatment decision of the malignant cells define the goal of the diagnosis of malignant diseases (Weller et al., 2014). In spite of increasingly improved morphological examination techniques, immuno-phenotyping, chromosome examination and other methods, significant differences in prognosis and response to therapy cannot ultimately be explained by the methods mentioned (Wen and Kesari, 2008). Therefore, to obtain more accurate information about the disturbed growth of tumour cells, explorations are being dedicated to the elementary building blocks of cells, the genes (Weller et al., 2014).

It is of particular interest to which genes in the tumour cells are active (turned on, expressed, high concentration of RNA) and which are inactive (switched off, not expressed, low or no RNA concentration) (DeRisi et al., 1997). So far, the expression of individual genes in cells after separation of RNA in the electric field on a gel using radioactively labeled probes was measured (Northern Blot). This process is very laborious and time consuming, and the experiments can only be carried out in special laboratories suitable for working with radioactive materials. In an experiment, only the analysis of a gene can be done (Schena et al., 1995).

In the early 1990s, the polymerase chain reaction (PCR) was developed. With artificially generated complementary DNA, the RNA can be repeatedly amplified. This makes it possible to detect even minimal amounts of RNA. In recent years, the technique of PCR has improved so much that it can also be used in clinical routine (for example, to detect viral RNA) (Livak and Schmittgen, 2001).

Microarrays

Analysis of microarrays

In the 1990s, so-called gene arrays were developed that allow the simultaneous expression analysis of many genes. First, it was possible to probe up to several hinder genes in parallel with gene probes attached to membranes. Since 1997, the technical prerequisites exist for the production of microarrays in which up to 30,000 gene probes can be applied to a glass surface for expression analysis (Schena, 2003).

The starting material for global gene expression analysis can be cells from tissue, blood, bone marrow or cell cultures. First, the total RNA is extracted and labeled with a fluorescent dye. Subsequently, the specific binding (hybridisation) of this labeled RNA takes place on the microarray (Livak and Schmittgen, 2001). On a solid surface of about 2 cm^2 (plastic membrane, glass surface or silicon surface) there are placed between 100 and 40,000 gene-specific probes to which the RNA to be measured binds. Subsequently, unbound RNA is washed off and the fluorescence for each gene-specific probe is measured with a high-resolution scanner (Livak and Schmittgen, 2001). Currently, two fundamentally different techniques are used:

1. In the cDNA technique, long cDNA fragments (500 to 5000 bp) are applied to the surface of the chip. From this technique derives the original name "DNA Chip", which is somewhat misleading because the gene expression measurement is carried out on RNA samples. Each cDNA fragment is specific for a gene. The RNA to be measured binds to the cDNA; The amount of bound RNA is proportional to the amount of RNA in the cells to be examined. In this technique, the hybridisation of a mixture of unknown RNA and a control RNA labeled with different fluorescent substances is performed. The quantification of the expression of the RNA in the target cells is done by measuring the mixed color and the intensity (Schena et al., 1995).

2. In contrast to the cDNA technique, the oligonucleotide microarrays use 25 to 80 bp oligonucleotides (so-called 25-mers or 80-mers) as gene-specific probes. The fluorescently-labeled RNA is hybridised directly to the microarray. Scanning with the scanner does not produce a color image but a black and white image with a corresponding gradation of the intensity of the individual measuring

points (Schena et al., 1995).

The numerical expression data of several thousand genes provide the opportunity to gain insights into the regulation of gene expression to an unprecedented extent. This flood of data is also a significant problem for the analysis. Initially, data analysis was done using simple spreadsheet programs. This made it possible to use the gene bank number to determine the expression of individual genes in different samples and to compare this expression virtually manually. Furthermore, pairwise comparisons could be made by systematic arrangement of the data. The analysis of several samples at the same time was not possible, and the evaluation was very lengthy (DeRisi et al., 1997).

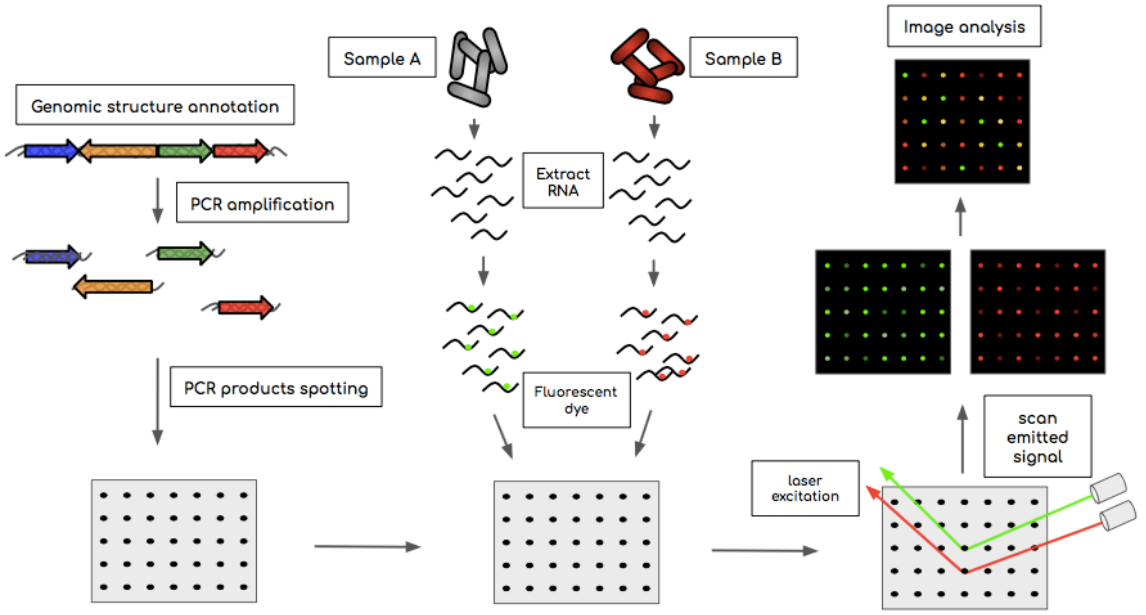

Figure 2.5: A diagram of a microarray analysis process.

Microarray Technology

In the last 10 years various powerful computer programs for evaluating microarray data have been developed. Simultaneously, various methods were introduced to statistically control the quality of the hybridisation data. Expression data can be graphically displayed and compared in pairs. Likewise, group comparisons are possible, e.g. Data from a normal control group with those from patients in the early or late stage of a disease (Allison et al., 2006).

Noise is one of the major problems associated with microarrays as it is introduced at each stage of the experiment. Since microarrays are noisy and an experiment is repeated more than once, using the same materials and preparations as in the previous experiment, many genes give different quantisation values due to noise after the sample and image processing steps (Miller and Tang, 2009).

Different mathematical algorithms can be applied to the gene expression data, with the help of which a characteristic gene expression profile can be created for each examined Prop. Groups of samples may be formed, e.g., represent a particular diagnosis, stage of disease or therapy success / failure (Miller and Tang, 2009). First, known samples are analysed. The program seeks to find commonalities in the gene expression profiles of samples with equal group affiliation. These form the so-called "learning set". Now samples with unknown group affiliation can be analysed (Schena, 2003). The program compares each individual gene expression profile of the unknown samples with the profiles that were created on the "learning set" and tries to assign the unknown samples to the individual groups. This method is mainly used in the diagnosis of diseases, in their prognosis estimation and in the determination of drug resistance (Schena, 2003).

When the first microarray experiments were performed on patient material, it was believed that the molecular genetic causes of cancer could soon be elucidated. In individual tumours, it has also been possible to discover cell biology relationships in the tumour cells. However, the hopes that microarray technology can quickly and systematically elucidate the pathophysiology of tumours have not yet been fulfilled. Probably the most significant role of microarray technology in the future lies in the classification and prognostic assessment of diseases based on gene expression. Much of the work with microarrays is therefore focused on linking diseases to specific expression profiles and using that information to validate existing or more accurate and accurate classifications (Heller, 2002).

RNA-Seq

RNA-Seq Analysis

RNA-Seq is a method of high throughput sequencing of cDNA transcribed fragmented RNA. In theory, RNA-Seq quantifies all transcripts present in a sample and increasingly displaces gene expression analysis by microarray. In conventional RNA-Seq, the entire transcript is fragmented and ideally each fragment is sequenced (Wang et al., 2009). The longer a transcript is, the more fragments are formed, so that long transcripts are over-represented in RNA-Seq results. This effect is compensated by a bioinformatic normalisation of the RNA-Seq data. However, different normalisation strategies lead to different results. In addition, it must be very deeply sequenced to be able to count rare and short transcripts with sufficient certainty. Another problem is the different RNA quality of samples. Once the RNA is degraded, normalisation becomes impossible (Wang et al., 2009).

The techniques for sequencing genomes and transcriptomes are very similar. First, the DNA or RNA molecules to be sequenced are fragmented and filtered according to their length. From the totality of the resulting fragments, a sequencing library is created by appropriate modification and duplication. Using sequencing machines based on the sequencing-by-synthesis principle, these libraries can be read within a few days (Kircher et al., 2009). For example, the Illumina Genome Analyzer II (Illumina Inc.) produces 100 to 200 million sequence fragments with a length of 40 to 200 nucleotides (Kircher et al., 2009). This is nearly ten times the size of the human genome sequence - but the cost of doing so is only ten thousands of the cost needed to sequence the first human genome some 20 years ago (Kircher et al., 2009). From this wealth of data, very accurate information about the investigated genome or transcriptome can be derived.

However, the new sequencing technologies still have some drawbacks despite their merits. It may e.g. not complete RNA molecules but only fragments are sequenced, and the sequencing error rate is well above the traditional methods (Morozova et al., 2009). Furthermore, both in the preparation and in the actual sequencing biochemical processes involved, which can change the concentration of RNA fragments undesirable. These properties make it much more difficult to utilise the resulting

sequences for the reconstruction and quantification of the transcriptome(Morozova et al., 2009). In addition, the amount of data produced during sequencing poses a very great challenge for the subsequent analyses. Previous methods for processing the resulting sequences are very easily reaching their limits, both with regard to accuracy and the speed of data processing (Morozova et al., 2009).

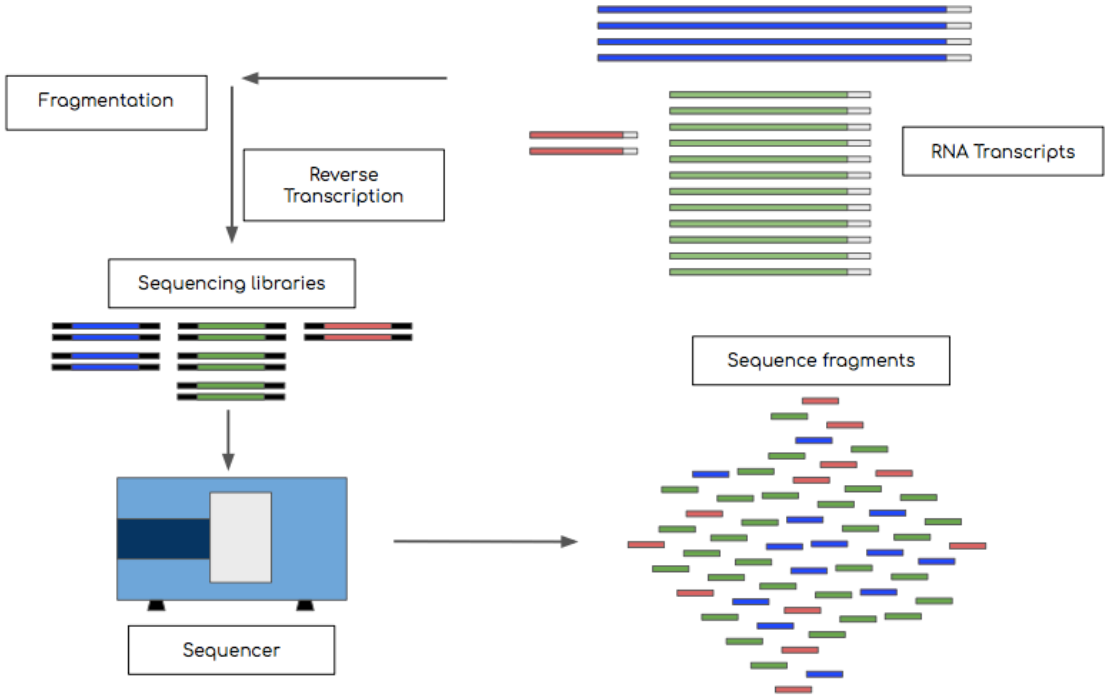

Figure 2.6: A diagram of RNA-Seq analysis process.

The genomes and transcriptomes are generally analysed using so-called "machine learning" (Libbrecht and Noble, 2015). This research field combines methods from artificial intelligence, statistics and mathematical optimisation. Machine learning deals with the analysis of complex statistical phenomena, such as the processing of RNA transcripts in the cell. For this purpose, empirical observations, the so-called learning samples, are analysed in order to be able to make precise predictions about the investigated phenomenon. Frequently, the exact and efficient core-based learning algorithms are used, which can be easily adapted to the respective problem by means of a so-called core function. These methods have been developed in research to the extent that they are now also suitable for the analysis of genome and transcriptome data (Libbrecht and Noble, 2015).

An important step in the investigation of transcriptome data is the quantification of the investigated transcripts in order to find volume-specific differences within the transcriptome or to compare different transcriptomes (Libbrecht and Noble, 2015). This may lead to deviations in the actual molecule concentrations due to molecular-biological preparation steps before sequencing. To account for these distortions in quantification, a new approach based on an optimisation approach has been developed that allows a much more accurate determination of the concentration of mixtures of co-occurring RNA transcripts (Tarazona et al., 2011).

Microarray VS. RNA-Seq

Understanding the regulation of gene expression is crucial to the knowledge of the correlation between genotype and phenotype. The requirements for an appropriate evaluation of transcription frequencies in samples has led scientist to discover and develop new technologies in gene expression profiling methods such as microarray and RNA-Seq. Questions are often being asked for which method is more practical in performing gene expression profiling, many researchers require difficult conclusion in terms of cost and performance value, with the additional impact of selecting the best method based on their research objectives. The following chapter elaborates the methods based on which os more practical in which context, while in some cases, both method will probably complement one another for its both advantage.

There are different viewpoints to either conclude which preferable gene expression profiling method, that require a discrete definition. First, practical questions such as the necessary genome information with the scope of finding the gene of interest. Another important question is the amount of the available resource of expertise required. Thus, financial contribution is also one of important factors to be considered for choosing the best method. Before, comparing the methods, similarities between the both has to be clearly defined.

Considerations

Equally as important to the aforementioned factors are the current research objectives of the project. In addition to measuring differences in gene expression between samples, accurate absolute Quantification is as well as important. Furthermore, the importance in discovering new genes is also worth mentioning, due to its cruciality to distinguish isoforms and the difference in expression between those isoforms from already existing genes. Thus, the interest at the expression of transcripts at very high or low levels or structural information such as alternative splicing and/or gene fusions should also included. After these objectives being undergone, the next steps is to find similarities between both method.

Characteristics

Microarray is a reliable method that has proven itself for the last few decades. Over time, most of the researchers became more and more familiar with the technology and analysis of gene expression results. Although, given as a previous issue, there is a general agreement on the basic processing methods that can now be implemented on any computer. Although the price of RNA-Seq have been reasonably reduced, microarrays are still more economical and provide higher workloads that offer great benefits when dealing with large-scale projects with larger sample quantities (Zhao et al., 2014). Microarrays are constructed with hybridisation probes that rely on sequence knowledge. Therefore, they cannot recognise structural variations of new genes or transcripts. The hybridisation strategy for microarrays limits their sensitivity, meaning that they cannot recognise the difference in expression between very similar sequences such as isoforms. In addition, they can only produce expression levels, not absolute quantification levels (Mooney et al., 2013).

Due to its independency on any prior sequence knowledge, RNA-Seq provides a complete overview of the transcriptome. Every single transcript, either known or unknown is sequenced in the sample. For this reason, structural variations such as new genes or transcripts, gene fusion and alternative splicing events are being able to be identified. RNA-Seq data be reanalysed as new discoveries become available, comparing to the microarray which has be run the analysis again to analyse the new

sequence information (Zhao et al., 2014). RNA-Seq, unlike microarrays that measure probe intensities, quantifies discrete digital read counts aligned to a particular sequence. Due to the individually sequenced transcripts, the method itself is more sensitive and more suitable for detecting low abundance and distinguishable of the biologically critical isoforms (Willenbrock et al., 2009). The dynamic range, in fact, can be adjusted unlimitedly through continuous sequencing. Since RNA-Seq is an advanced technology, it is a fundamental analysis method to most researchers. One of the bigger drawbacks is the data output, which be much more complex compare to microarray, which can lead to a more much complex interpretation of the analysis results (Mooney et al., 2013). In fact, the specific computer infrastructure and personell must use the additional biological information obtained in these data. Since various RNA-Seq analysis tools are rapidly evolving, there are yet no standard protocol procedures provided, which can make it difficult to compare results. Indeed, RNA-Seq generates so much data that storing such larges datasets can be a big problem. Large data sizes can be very difficult to access and also be very expensive to store, especially for large projects with a larger number of samples (Kogenaru et al., 2012).

Finally, despite the reduction of RNA-Seq costs which are based on modern technology, the cost of performing a microarray experiment is in most cases lower than in a comparable experiment based on RNA-Seq. In many cases, a combination of technology that maximises efficacy and overcome limitations can be the best strategy. For example, RNA-Seq results can largely cover all transcripts of samples of different temporal and spatial origin, without the need for sequence knowledge. Because of the relatively higher cost and throughput of sample handling and data analysis, systematic tracking of sequencing results for validation and/or profiling can be reproducibly long and expensive in repetitive sequence. Customised microarrays can be effective follow-up tools for capturing comprehensive sequencing information and profiling and comparing gene expressions quickly, reproducibly, and cost-effectively based on sequencing errors.

Outlook

The global gene expression analysis of malignant cells, but not only of these, will in the future gain substantial importance in basic research and increasingly in clinical medicine. This makes it possible to record the functional states of the genes of a cell and thus risk profiles, disease progression and therapy response. Since one cannot generally assume that the change of only one gene is specific to the disease process, the diagnosis and prognosis of diseases will be based on certain gene expression profiles characteristic of the particular disease or clinical course (Schulze and Downward, 2001).

Limiting for the microarray analyses is the relatively large amount of RNA (about $10\mu g$), which is needed for a hybridisation. So far, this cannot be applied to a small number of malignant cells in a tissue sample (Drăghici, 2016). However, there are already approaches on how the sensitivity of the technique can be increased. The ultimate goal could be the analysis of gene expression in a single cell. An advantage of the method is that it will be largely automatable (Miller and Tang, 2009).

The experience to date shows that the reproducibility of the results from microarray experiments strongly depends on the quality of the RNA. When applying the technique in basic research, this point is not a problem, but for the investigation of clinical samples. The development of useful RNA preservatives, which are added immediately after the cell or tissue sample has been taken, seeks to solve this problem with regard to the clinical application of the method (Schena et al., 1995).

The technique of microarrays and the methods for analysing the expression data will be improved in a few years so that their broad application in the laboratory sector is standard. The risk of uncritical accumulation of gene expression data, which then confront the attending physician or the patient (Drăghici, 2016). It should always be the goal to make a critical selection of the genes to be studied. It remains a vision of the future, with the microarray technique not only to find the actual state of a cell for diagnosis, prognosis and treatment decision, but also to identify individual disease responsible genes. If it were possible to detect the function of these genes or their malfunction, for example in signal transduction, then causal, targeted molecular therapies could be developed (Schena et al., 1995).

Since microarray experiments can be processed differently, there is no consistent pro-

cedure in evaluating microarray data. Also, the order of the process varies depending on application. Furthermore, an error analysis is also carried out. Statistical analysis requires different evaluations from different fields between biologist, computational statisticians, and molecular and clinical research to development more statistical methods for solving problems associated with microarray analysis. This has led to the development of different comprehensive statistical approaches (Drăghici, 2016).

The application of the new sequencing technologies and the newly developed methods make it possible to generate very accurate images of transcriptomes in the computer and to determine their change under different experimental conditions (Libbrecht and Noble, 2015). Exact knowledge of the transcriptome also allows the application of learning methods that can learn from the measured data how the transcriptome is formed from the genome and other factors. It has also been shown that epigenetic information and external influences also strongly influence the transcriptome (Tarazona et al., 2011).

Experimental data

The data was retrieved from The Cancer Genome Atlas (TCGA) through a pipeline that was established by my supervisor Prof. Peter Sykacek with gene expression annotation file, where all the clinical information are included within this file, and all microarray raw files from Affymetrix .CEL files and from Agilent .txt files. The Cancer Genome Atlas was created in collaboration of The National Cancer Institute (NCI) and National Human Genome Research Institute (NHGRI) for creating global and complex genomic maps from different cancer types. The cancer data are available publicly and were utilised and processed for different research studies by researcher globally (Tomczak et al., 2015)).

In addition, The Cancer Genome Atlas has developed a pipeline of genomic data analysis that can accurately assemble, select and evaluate human tissue for large-scale genomic compounds. The success of the TCGA project has had an impact on teamwork in science and can serve as a prototype for future projects. The Cancer Genomics Center (CCG) is an NCI initiative that will replace TCGA and builds on the success of TCGA by publishing genomic data using a similar collaboration for a

complete genomic analysis (Tomczak et al., 2015).

The Affymetrix raw files was in total of about 2.4 GB and the Agilent raw files about 171 GB. After preprocessing the microarray raw files, the preprocessed data matrix of Affymetrix has 200102 rows (genes) and 536 columns (samples) and that of Agilent has around 13948 rows (genes) and 536 columns (samples). The HTSeq-counts and FPKM files have a total of only 153 patient samples. All retrieved gene expression data were preprocessed using specific methods provided by Bioconductor (Gentleman et al., 2004).

Once the all the data from each experiments have been converted into a data frame, they were cross annotated by matching and filter all the features/genes that occur in all 3 experiments and sorted them in equal order. The same procedure were also performed by the samples.

Description of the GDC Data Portal Webpage

The GDC Portal is the TCGA GDC Webpage for retrieving GDC data. The GDC portal allows to search for data provided in the GDC database. It is a user friendly graphical interface. You can search and browse all patients and their corresponding metadata. The GDC portal contains different connections that can be linked to each other and be able to specifically retrieved required data for further analysis (Network and others, 2017a).

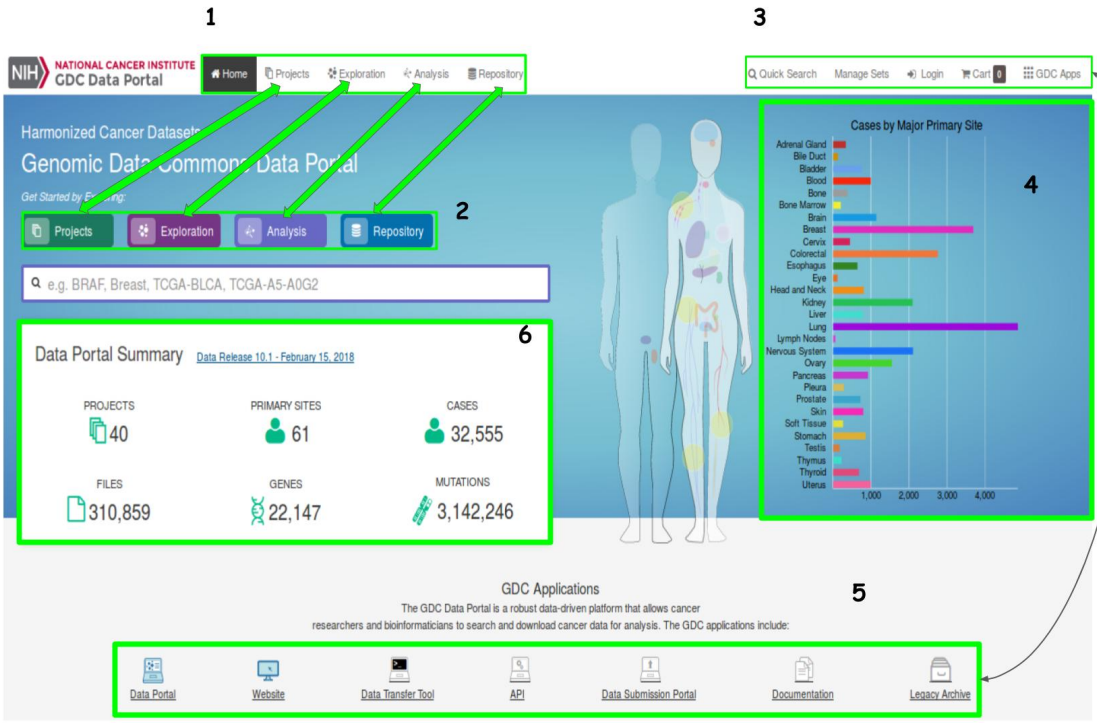

Figure 2.7: Graphical Interface of the GDC Data portal webpage(Network and others, 2017a).

1. The left main menu bar. These are the quick links to the GDC data contents to Projects and Repository. At the Exploration menu, analysis using GDC's DAVE tools can be found (GDC Data Portal User Guide 2018).
2. Buttons identical to the left main menu bar that leads to the same Projects, Repository and Exploratory views as the main menu (Network and others, 2017b).
3. The right menu bar, Search, Login and View Cart can view the right main menu. The GDC Apps is a link to the whole section of the GDC Webpage, including the access to the legacy archive, to the cBioPortal and to the documentations (Network and others, 2017b).
4. The bar chart connected with the human model next to the left that can bee seen above the overview figure. These are the cases for each part of the human body and organs that included in the GDC pipeline. By clicking on one of the bar-graphs, it will take you to the Project menu for the selected project (Network and others, 2017b).
5. A much larger view to the GDC Apps links, which contain the same links as

in the right menu bar (Network and others, 2017b).

6. The Data Portal summary menu are shown the total number of projects, cases, primary sites, files, genes and mutations annotations that are currently stores in the GDC pipeline. These numbers changes from time to time, since new data als always added (Network and others, 2017b).

GDC Legacy

GDC Legacy Archive is where the original data in GDC the call it the legacy data that was provided by the original submitter that used old genome build data. These legacy data are unharmonised und not-maintained data by the GDC. The Legacy Archive can be accessed at the GDC Portal under the GDC Apps links and by clicking Legacy archive.

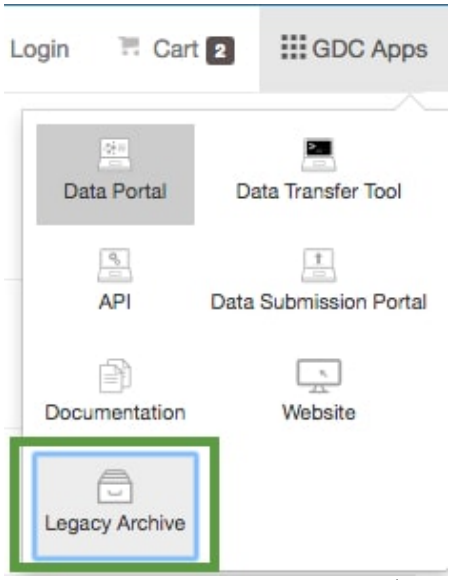

Figure 2.8: Accessing Legacy Archive from GDC Apps(Network and others, 2017a).

The GDC Legacy archive consist of features based on the GDC Portal Projects menu bar, where all the GDC unharmonised legacy data content are shown. The left panel are facets that contain filters for specific search and the right panel shows the results of specific searching the left panel to the legacy data. The difference between the GDC Portal and the Legacy archive are that there are no pie charts representing

the visualisation of the Data content in the legacy archive. The File and Annotation tables is a list containing all of the legacy files and annotations. The GDC Data Portal and Legacy Archive are not connected together, they are separated systems which is being recommended that Legacy users should use the GDC Portal instead due to harmonised and maintained (Network and others, 2017a).

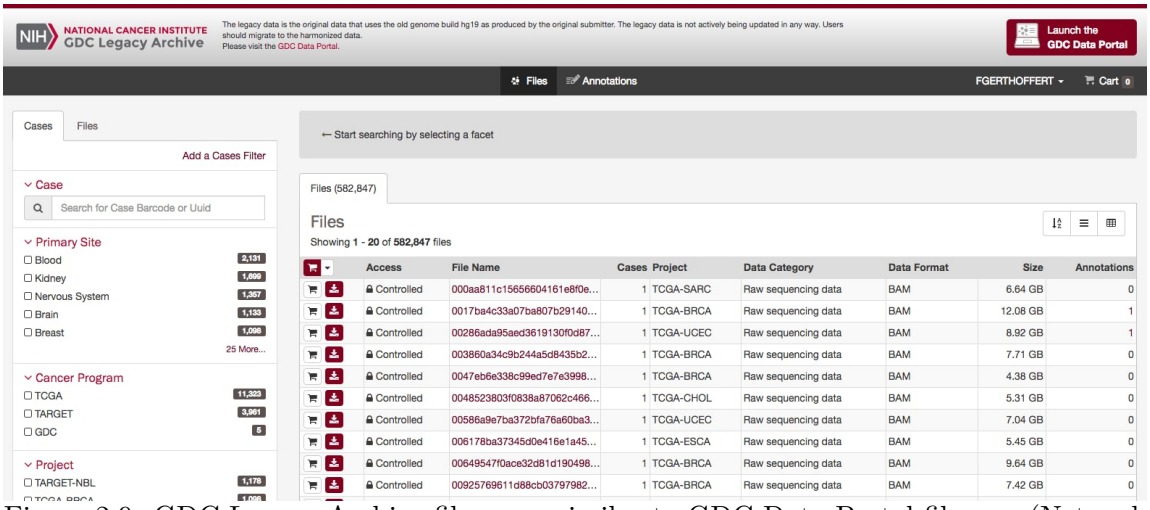

Figure 2.9: GDC Legacy Archive file page, similar to GDC Data Portal file page(Network and others, 2017a).

The file page view of the GDC Legacy Archive shares identical features with the GDC Data Portal file page view. The main difference between both are that GDC Legacy Archive provide additional information as describe with the figure below (Network and others, 2017a).

Materials used in Data Analysis

The major languages used in data analysis of TCGA microarray data are python and R with their corresponding packages from Bioconductor, Sklearn, Pandas, Numpy, Matplotlib, Scipy, Statsmodels.

Python

The programming language Python was developed in the early 90's by Guido van Rossum at the Stichting Mathematisch Centrum. It was named after the BBC show Monty Python's Flying Circus. Since then, the language has undergone numerous changes and is now available in version 3.6. Python was thought to be a redesign of the learning language ABC and the basis of the operating system Amoeba (Beebe, 2018).

The goal of Python is a simple and clear language, with enormous functionality and only a few keywords. The focus was still on short development times and limiting the programmer only as much as is absolutely necessary (Beebe, 2018).

Since 2001, Python has been management, published and promoted by the Python Software Foundation (PSF). It is a non-profit organisation supported by sponsors from a variety of fields (Beebe, 2018).

Python is a typical high-level language that allows very abstract programming. The user does not need to worry about low-level problems and has a large amount of complex commands at his disposal. Data types such as complex numbers, strings, tuples, lists, and dictionaries are already implemented as standard data types and can be easily used (Beebe, 2018).

Like many other languages, Python is interpreted. This is possible in two ways: The source code can be sent precompiled to the interpreter for execution, or as source code, which then has to be internally compiled first (Beebe, 2018).

Like any newer language, Python is object oriented. This does not mean the you are required to work with classes. As with C++, and unlike, for example, Java, Python also makes possible to declare functions and variables outside of classes. The key words private, protected and public, which can be found in most other object-oriented languages, are completely omitted and left to the programmer. To mark methods private, its name must begin with two underscores. By name-mangling this is the hidden, but still accessible from the outside. Since Python is already an interpreted language, it has been designed to run on as many platforms as possible (Beebe, 2018).

In contrast to most other languages, Python does not use its own key words or

symbols (such as Begin and End in Pascal) for statement grouping, but only the indentation. Whitespace at the beginning of a line is therefore assigned a meaning. As an indication, a tab is generally equated with eight spaces. Therefore, it makes sense to indent within a module only with spaces or only with tab. Otherwise, it can easily lead to errors in the block formation (Beebe, 2018).

In general, Python is syntactically and programmatically very elegant. The statements are relatively short, unique and never need to be completed with a semicolon or dot as usual. Added to this is block formation through indentation, which to some extent imposes a uniform look on many programs (Beebe, 2018).

Python libraries

For our clustering analysis we are applying different kinds of python libraries but mostly sklearn for the clustering analysis(Pedregosa et al., 2011), pandas for data manipulation, Matplotlib for data visualisation (Hunter, 2007), Numpy for multi-dimensional array and matrix manipulations (Walt et al., 2011) and scipy for statistical analysis (Jones et al., 2014).

Scikit-learn or also Sklearn is a free Python library dedicated to machine learning. It is developed by many contributors, particularly in the academic world by French institutes of higher education and research. It includes functions for estimating random forests, logistic regressions, classification and clustering algorithms, It is designed to harmonised with other Python free libraries, including Numpy, Pandas and SciPy (Pedregosa et al., 2011, Walt et al. (2011), McKinney (2015)).

The Pandas we are writing about in this chapter have nothing to do with the cute panda bears. Pandas is a Python module that rounds off the possibilities of Numpy, Scipy and Matplotlib. The word Pandas is an acronym and is derived from “Python and Data Analysis” and “Panal Data” (McKinney, 2015).

There is often confusion about whether Pandas is not an alternative to Numpy, Scipy and Matplotlib. The truth is that Pandas is building on Numpy. This also means that Numpy is for Pandas prejudice (Walt et al., 2011). Scipy and Matplotlib are not required by pandas but are extremely useful. Therefore, the Pandas project also lists these as “optional dependencies”.

Pandas is a software library written for Python. It is used for data manipulation and analysis. It provides special functions and data structures for the manipulation of numerical tables and time series. Pandas is a free software and was released under the three-clause BSD license (McKinney, 2015).

NumPy is an acronym for "Numeric Python" or "Numerical Python". This module is an open source extension for Python that provides fast precompiled functions for math and numeric routines. In addition, NumPy enriches the Python programming language with powerful data structures for efficient arithmetic on large arrays and matrices. The implementation even targets extremely large ("Big Data") matrices and arrays. Furthermore, the module offers a huge number of high-quality mathematical functions to work these matrices and arrays (Walt et al., 2011).

SciPy (Scientific Python) is often called in the same breath as NumPy. SciPy extends the power of NumPy with additional useful features such as minimisation, regression, Fourier transformation, and many more (Jones et al., 2014).

Matplotlib is a library for plotting like GNUplot. The main advantage over GNUplot is the fact that Matplotlib is a Python module. Due to the growing interest in the Python programming language, that popularity of Matplotlib is also increasing. Matplotlib is able to create diagrams and representations in different formats, which you can then use in publications (Hunter, 2007).

R programing software

R stands for the R Project for Statistical Computing.

- R is a software for statistical data processing and its graphical visualisations.
- It is an implementation of the statistical programming language S that runs on various UNIX, Linux and Unix-like operating systems, as well as on Windows and Mac OS X.
- Older R versions are still available for the classic Mac OS.
- Many operating systems already have compiled packages.
- The language can easily be extended by new functions and a large number of existing additional packages supplement the R functionality with methods from the special and application areas of statistics.

- R can be connected to other programming languages such as Perl, Python, C or Java.
- Furthermore, R can be used both interactively, in single command mode, as scripting language and in batch mode. The R source code is published under the GNU General Public License (GPL) of the Free Software Foundation.

R was created in 1997 as an open source alternative for the then extended S-PLUS business statistics software, whose programming language "S" is modelled by R. R is included in the language of the so-called basis of "environmental statistics" (as the opposite of "statistical software"). This should highlight R. R's open source concept consists of a few major packages that provide basic functionality, and can be expanded with any number of packages. While most of these packages are also available under an open source license, the R license also allows you to offer commercially licensed extension packages. In other part, the additional development of R is also funded by such packages, e.g. be developed as a work commissioned for the pharmaceutical industry (R Development Core Team, 2008).

In addition to the popular statistical analysis programs such as "SPSS" or "STATA", R has the advantage of being available for free (under the free GNU license) around the globe. R can import most common formats, ensuring full control over the data and providing a reliable, open-source format for created datasets. In addition, R is partially more powerful and there are more evaluation methods available than other programs (R Development Core Team, 2008).

R is a programming environment. Functions can be easily adapted to its own needs. Complex problems can be solved even if the developers have not yet implemented them. R is being continuously developed and expanded by the scientific community. New statistical methods are usually integrated in R. A standardised package system facilitates the subsequent package system as well as the publication of own packages. R is also a working user and developer community that is open to questions, making it easy to get started. R can be used across systems on different platforms, has highly flexible interfaces for data input and output and can work with several other applications (R Development Core Team, 2008).

After listing all the advantages, there are some difficulties when using R, for the beginner, the operation functionality of R is not in need of getting used to. When

programming in R, compared to other modern languages, some things work in unexpected ways and certain basic methods are currently only cumbersome or not implemented at all (R Development Core Team, 2008).

Bioconductor

Bioconductor is an open source development software project, that provides tools for high throughput genomic data analysis using R. Additionally, it contains different packages which supports annotations from different fields. In addition, Bioconductor software development projects desire is to provide publicly accessible statistical and graphical methods for genomic data analysis, assisting researchers proliferate scientific findings on computational methods applied in genomic data analysis (Gentleman et al., 2004).

Analysis Overview

The Figure 2.10 represents overview of the data analysis approach as a flowchart. The workflow is composed of 3 main parts: data preparation, clustering analysis and biological interpretation. The flowchart visualises the importance of data preparation, since the clustering results of gene expression data rely on how the data are prepared.

A background correction has to be performed in able to extract noises due to specific hybridisations (e.g. detecting signal which do not occur from the hybridised probe samples) (Simon et al., 2003). The goal of a gene expression data preparation is to obtain a high quality of intensity value that can be contemplated proportionally to the expression level. A part of the intensity value comes from the non-bound probes (e.g. a small amount of sample may interact to the non-complementary chains), as well as signals from unknown or unwanted sources. Removing those unwanted signals from the sample probes are the main task of a background correction (Simon et al., 2003).

Which much effort and resources has been dedicated in developing different methods, normalisation of raw data, which is responsible of the regulation between the

technical variation between arrays, is one of the essential key-point in data preparation (Gentleman et al., 2006). Leaving the biological variation unmodified while extracting out as much of the noises and variations as possible is one of the most challenging tasks of gene expression data normalisation. These challenges makes the most amount of effort, as these are only being part of the main issues in data preparation. First, a visual comparison of the raw against the preprocessed data is a crucial part of data quality assessment for choosing the most suitable normalisation method to estimate the normalisation performance (Gentleman et al., 2006).

Quality control is an essential part of the preprocessing gene expression data, to determine the reliability of the data preprocessing. Quality assessment is a primary concern of gene expression data preprocessing. The main goal of quality assessment is to identify outlier arrays and calculates the signal-to-noise ratio.

One of the main purpose of clustering gene expression data is to identify the regulated biological processes by evaluating co-regulated genes, based on the question that the cellular response is mainly reflected by the transcription levels. Unfortunately, the assignment of genetical co-regulation and biological function is not the same. The reasons are diverse, mainly due to a variety of biological responses. First of all, cellular processes are affected by the adjustment of up- and down-regulation. Therefore, the genes involved in common pathway can reach completely different groups. Secondly, post-translation modifications regulate many biological processes. In particular, statistical fluctuations makes the clustering more inaccurate. The less statistical variation throughout the data, the more robust are the clustering algorithms. To form meaningful clusters, a minimum number of samples are required. The potential for genes to clusters into groups with meaningful biological outcomes has been implied on many clustering algorithms, with the fact that the algorithms perform differently on the same datasets.

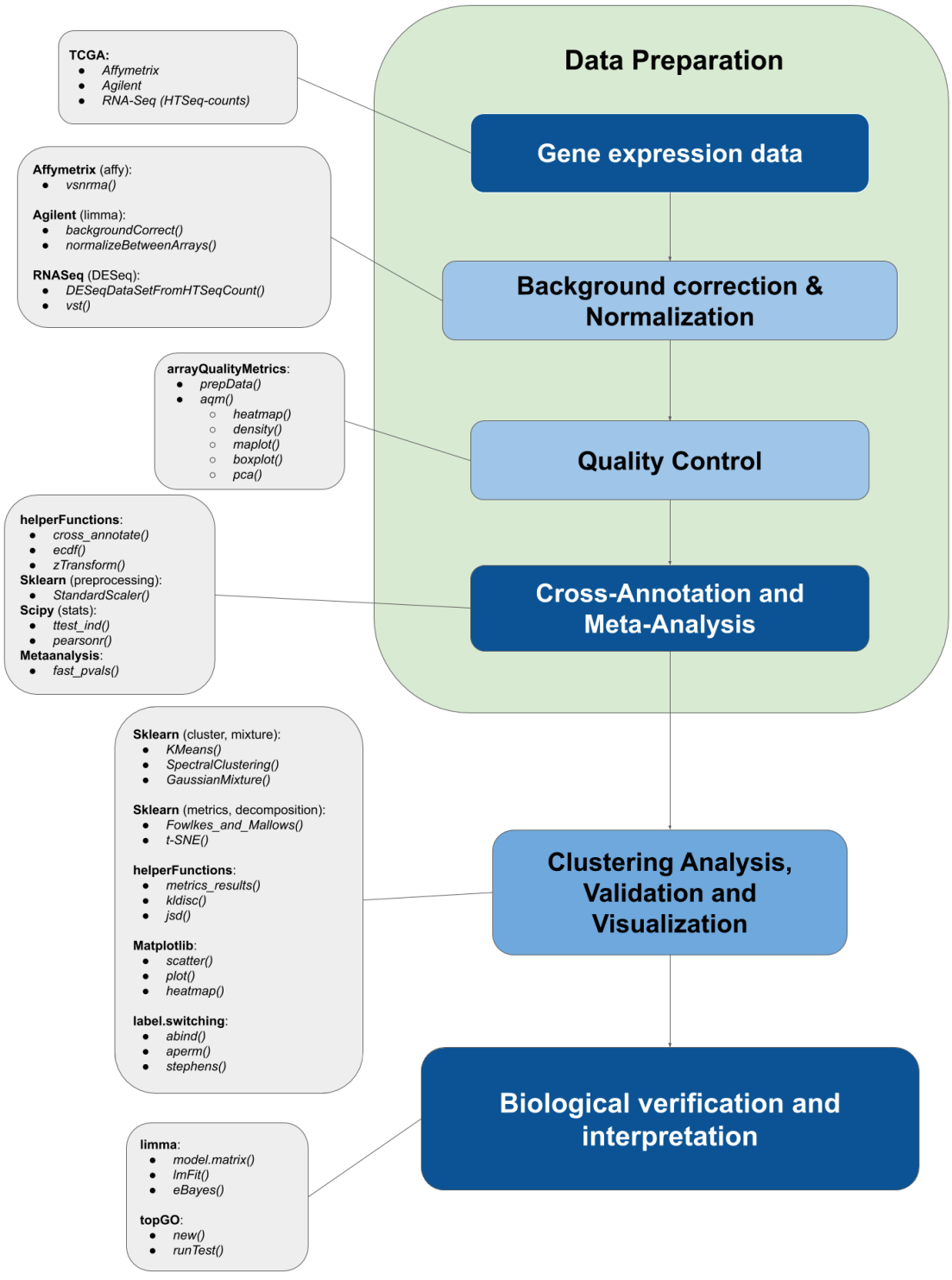

Figure 2.10: Overview of statistical gene expression data analysis approach.

Preprocessing and Quality assessment

The scripts were created with R-Studio. The scripts use packages developed for the evaluation of cDNA microarray data. The main package for the analysis is Bioconductor (http://www.bioconductor.org/). This is a free package for R designed to analyse and compare genomic data. The package was designed in the fall of 2001 and has been receiving regular updates ever since. It supports a large number of biological analysis functions, including analyses for affymetrix arrays, genome annotation and extensive graphical analysis(Gentleman et al., 2004). All packages used and their meaning can be seen in the table below.

library	defintion
limma	Linear models for the evaluation of geneexpression data.
affy	Basis for the analysis of Affymetrix microarrays.
DESeq2	Basis for the analysis of RNA-Seq data.
genefilter	Filter for better search for differentially expressed genes.
vsn	Variance stabilization and calibration for microarray data.
arrayQualityMetrics	Various quality assessments for microarray data.

Affymetrix microarray data

Affymetrix was preprocessed with the *affy* package provided(Irizarry et al., 2006). In able to read and to parse the raw .CEL files, it is necessary to put the Affymetrix file names as a list. Using the *ReadAffy()* function, the list with the file names is being iterated that reads each files located within a specific directory, which result an affy specific object.

Removing outliers was done with a similar approach as Agilent, by preprocessing the data and detecting outliers using the *aqm()* function. Preprocessing Affymetrix was done with the *vsnrma()* package, similar to the *normaliseVSN()* of Agilent, background correction and normalisation without reading and parsing the .CEL files. As the result, an expression set was generated, which is then utilised for detecting outliers using *aqm.boxplot()*, *aqm.density()*, *aqm.heatmap()* and *aqm.maplot()*. After removing the first outliers, the process was also re-run until no outliers could be

detected.

Agilent microarray data

Agilent was preprocessed using *Bioconductor Limma* package(Smyth, 2005). The *read.maimages()* function reads and parse raw Agilent files as a .txt file into a limma specific object similar to an expression set, which the file names are required to be in a list to generate the limma object.

For removing outliers, *backgroundCorrect()* and *normalizeBetweenArrays* was being utilised for the normalization with the corresponding quality assessment using the *prepdata()* function from *arrayQualityMetrics* library with *do.logtransform* set to TRUE. For outlier detection, we used 4 different methods using *aqm()* included in the *arrayQualityMetrics* package, which are *aqm.boxplot()*, *aqm.density()*, *aqm.heatmap()* and *aqm.maplot()*. After removing the first outliers, the process was re-run until no outliers could be detected. After outliers removal, Agilent was preprocessed using *Bioconductor Limma* (Smyth, 2005) using *normalizeVSN()*, which included background correction and normalisation. The function returns an *MAList* on a log2 scale.

RNA-Seq data

2. For the RNA-Seq data, we obtained raw counts from HTSeq data. *DESeq2* library was being used for preprocessing the data. *DSEqDataSetFromHTSeqCount()* function provides the function to parse and merge the data into an array(Anders and Huber, 2010). The *varianceStabilizingTransformation()* function calculates the variance-stabilizing transformation of the array, the purpose of transforming sample mean values variation, constant throughout the samples, this includes the calculation of the count size factors for the constant variance along the mean values, the *rlog()* for the logarithmic transformation of the data. Furthermore, quality control for the outlier detection were also applied to take the outlier samples out of the array.
3. For FPKM data, we merge the data into a data frame and transformed the data

with the corresponding phenotypic data and meta data, which corresponds to each samples and feature data, which annotates each probe sample ID. After transforming the data into an expression set, the function *justvsn()* from the *vsn* library were then used for further preprocessing.

The gene and the sample order were then subsequently sort and cross-annotated in the same order. All genes and samples that don't exist between either of the experiments have been remove from the dataframe so all experiments together share the same variables. Then the sample order were also sorted in the same order. After preprocessing the data, clustering analysis were performed that will be described from the next chapter.

Meta-Analysis

For the clustering analysis, low variable genes that were not consistent throughout the gene expression data from all experiments had to be excluded. The modified meta-analysis by P. Sykacek is an adaptation of the meta p-value calculation by the approach in combining dependent p-values with an empirical adaptation of Brown's method(Poole et al., 2016), filtering the genes were possible by combining p-values across the experiments.

The idea behind Meta-Analysis of Pathway Enrichment is to combine Independent and dependent omics data sets (Kaever et al., 2014) to obtain a variable specific data matrix which allows the calculation of covariance estimates by the approach in the empirical adaptation of Brown's method(Poole et al., 2016) and the analytic approximation in combining dependent p-values (Kost and McDermott, 2002).

First a function was written to calculate the meta p-values by the fisher method (Elston, 1991). The input required a matrix where the rows represent variables (e.g. expression of a particular gene) which were harmonised in such that all columns represent the same variable. The columns of the matrix represent the matrix represent different experiments which provide evidence for the same hypothesis. The result is a 1-dimensional column vector of aggregated meta p-values.

The next function required for the modified meta-analysis is a function to convert a

matrix of p-values into a 3-dimensional tensor of data matrices. The input required a matrix where the rows represent variables (e.g. expression of a particular gene) which were then harmonised, so that all columns represent the same variable. The columns of the matrix represent the matrix represent different experiments which provide evidence for the same hypothesis. The result is a 1-dimensional column vector of aggregated meta p-values. Another input was also to convert this matrix with the inverse cumulative density function (CDF) to give out the probability of the normal gaussian distribution. An odd number of values has to be generated for determining a variable specific covariance structure among experiments. The larger the number the larger the set of similar variables will be selected for modelling covariance. The larger the value the better the covariance estimate and the stronger the similarity assumption. As the result of this function a 3-dimensional tensor of p-value derived data samples which has been returned for calculating the covariance similarity structure.

This next function the purpose was to calculate the aggregated summary required for the previous function to calculate the moderated meta p-values according to Brown's method. We allowed for the empirical Brown method (W. Poole, 2016) and Kost approach (Kost and McDermont, 2002). As input it was the output from the previous function, the 3-dimensional tensor that calculates the covariance contribution and as the result a 2-dimensional matrix with summaries from the sample covariances required for Brown's method(Poole et al., 2016).

By combining these functions together we created a function that calculates meta p-values with the empirical brown method(Poole et al., 2016) and Kost method (Kost and McDermott, 2002) with the idea of Meta-Analysis of Pathway Enrichment (Kaever et al., 2014).

Cluster Analysis

The following task subsequently after calculating the meta p-values was filtering out the first 2000 genes that were significant after the meta analysis. Clustering analysis were then performed by different numbers of gene subsets. The first 50, 100, 500 and 1000 features was implemented with different clustering algorithms.

As for the cluster analysis, the definition is an unobserved method of grouping data. In contrast to the classification the true class labels are not known. Cluster analysis is performed using a cluster algorithm. The goal is that the objects within a cluster are very similar, while objects in different cluster are very different. The "natural" structure of the data should be revealed. Possible applications of cluster analysis include data reduction and the formation of hypotheses and predictions (Eisen et al., 1998) that are used in many fields: bioinformatics (Sturn et al., 2002), machine learning and Pattern Recognition (Nasrabadi, 2007).

The aim of the clustering is to find groups of similar observations (clusters), where similarities are detected by means of a similarity function, for example a distance measure (Ackermann et al., 2010). There are a very large number of different clustering methods (Estivill-Castro, 2002), which are categorised differently in the literature, e.g. (Tan et al., 2005, Soni and Ganatra (2012), Berkhin (2006), Rokach and Maimon (2005)). Of the many clustering principles, only those that are used in this work will be discussed here: hierarchical clustering, partitioning clusters and density-based clustering (Xu and Wunsch, 2005).

- In hierarchical clustering, it is assumed that clusters are nested hierarchically, e.g. two clusters together can form a higher-level third cluster. Hierarchical clustering techniques are divided into agglomerative and divisive hierarchical clustering (Aggarwal and Reddy, 2013). In agglomerative clustering, each point is treated as a single cluster at the beginning, combining at each step the cluster nearest to one cluster until, after a certain number of blocks, only one cluster remains - the complete data set. In divisive clustering, the hierarchical cluster structure is developed by starting with a single cluster - the complete data set - and dividing it until each point is a single cluster (Aggarwal and Reddy, 2013).
- With partitioning clustering methods, it is assumed that the clusters are completely separated from each other and no hierarchical structure exists. The goal of partitioning techniques is to find the best possible separation of the data into separate clusters (Aggarwal and Reddy, 2013).
- Density-based clustering methods follow the idea that clusters are

(a) regions in space where high densities prevail at points and (b) clusters are separated by areas in space where (significantly) lower densities prevail. What

exactly is “density”? There are several definitions for this term, whereby in this master thesis the centre-based concept of density (Xu and Wunsch, 2005) is used. The centre-based density is determined by an observation b, in which one counts the observations that are within a certain distance around

b. The more observations are within a certain distance of the observations b, the higher the density.

In this thesis, three cluster techniques were implemented: Gaussian Mixture Model (GMM), K-Means Clustering, and Spectral Clustering.

The procedures were chosen due to the modes of operation, approaches, strengths and weaknesses - and thus the data can be examined from different perspectives.

Gaussian Mixture Model

A Gaussian mixed model is a weighted sum of Gaussian distributions. Gaussian mixed models are used to model complex multimodal distribution functions. Multimodal means that the distribution has more than one maximum(Reynolds, 2015). The probability density function is given as:

$$p(x|\Theta) = \sum_{m=1}^{M} \alpha_m \mathcal{N}(x|\Theta),$$

in which $\Theta_m = \{\mu_m, \Sigma_m\}$. The parameters of the distribution are: $\Theta = \{\alpha_1, .., \alpha_M, \mu_1, .., \mu_M, \Sigma_1, .., \Sigma_M\}$, in which M is the number of gaussian components. The parameter $\alpha_m = P(m)$ weights the individual gaussian components. For the probability α_m applies $0 \leq \alpha_m \leq 1$ and $\Sigma_{m=1}^{M} \alpha_m = 1$. The scaling of $\mathcal{N}(x|\Theta)$ by α_m guarantees that $\int_{-\infty}^{\infty} p(x|\Theta)dx = 1$ (Bishop, 2012).

Maximum-Likelihood Estimator

Given the data $\mathcal{X} = \{x_1, ..., x_N\}$, the parameter of the model can be estimate with the ML-estimator. The goal is to estimate the Θ of the parametric model. In the maximum-likelihood method, the parameters Θ are estimated so that the likelihood

function becomes maximum, the data $\mathcal{X}$ is fixed and the estimated parameters Θ vary (Dempster et al., 1977).

The likelihood function is defined by:

$$P(\mathcal{X}|\Theta) = P(x_1, ..., x_N|\Theta) = P(x_1|\Theta)P(x_2|x_1, \Theta)....P(x_N|x_N - 1, ..., N_1, \Theta)$$

If the samples are independent and identically distributed, we can determine the likelihood function by the product of the likelihoods of all x_n(Dempster et al., 1977), that means:

$$P(\mathcal{X}|\Theta) = \prod_{n=1}^{N} P(x_n|\Theta)$$

can be used over the samples. This avoids numerical problems with very large N. The log likelihood function is:

$$L(\mathcal{X}|\Theta) = L(x_1, ..., x_N|\Theta) = lnP(x_1, ..., x_N|\Theta)$$

$$= \Big[\prod_{n=1}^{N} P(x_n|\Theta) \Big]$$

$$= \sum_{n=1}^{N} (ln(Px_n|\Theta))$$

In the maximum likelihood method, the parameters Θ with the maximum $L(X|\Theta)$ are of interest, that means, we need the argument Θ, which maximises the log-likelihood (Dempster et al., 1977):

$$\Theta_{ML} = \underset{\Theta}{argmax} L(\mathcal{X}|\Theta)$$

To determine these parameter Θ_{ML}, the log-likelihood function is derive to Θ and the derivative is set to 0 (Dempster et al., 1977).

$$\frac{\partial L(\mathcal{X}|\Theta)}{\partial \Theta} \stackrel{!}{=} 0$$

For example, in the case of a Gaussian distribution, the log-likelihood function is:

$$L(\mathcal{X}|\times = \sum_{n=1}^{N} ln(\mathcal{N}(x_n|\Theta))$$

By deriving the function above the parameters μ and Σ and zeroing the derivative (Hartley, 1958), we obtain the $\Theta_{ML} = \{\mu, \Sigma\}$ the following results (Dempster et al., 1977):

$$\mu = \frac{1}{N} \sum_{n=1}^{N} x_n$$

$$\Sigma = \frac{1}{N} \sum_{n=1}^{N} (x_n - \mu)(x_n - \mu)^T$$

Extimating the parameter Θ

The Gaussian mixed model is a parametric model. The estimation of the parameters O can be done by the maximum likelihood method (McLachlan and Krishnan, 2007). The given data are: $\mathcal{X} = \{x_1, ..., X_N\}, x \in \mathbb{R}^d$ and the parameters are: $\Theta_{ML} = \underset{\Theta}{argmax}\{ln\, P(\mathcal{X}|\Theta\}$

In the first step, the log-likelihood function $L(\mathcal{X}|\Theta) = lnP(\mathcal{X}|\Theta)$ is formulated for the gaussian mixture model (McLachlan and Krishnan, 2007). Then the stationary point for $L(\mathcal{X}|\Theta)$ is searched: $\frac{\partial lnP(\mathcal{X}|\Theta)}{\partial \Theta} \stackrel{!}{=} 0$

Assuming that the samples/data points $\mathcal{X} = \{x_1, ..., X_N\}$ are independent and identically distributed, the following log-likelihood function results (McLachlan and Krishnan, 2007):

$$L(\mathcal{X}|\Theta) = \sum_{n=1}^{N} lnP(\mathcal{X}|\Theta) = \sum_{n=1}^{N} ln \sum_{m=1}^{M} \alpha_m \mathcal{N}(x_n|\mu_m, \Sigma_m)$$

In the last step, the Gaussian mixture model is used for $P(x_n|\Theta)$. Next, the individual derivatives for the parameters $\mu_m, \Sigma_m, \alpha_m$ are formulated and set to 0 (McLachlan and Krishnan, 2007).

For the derivation of the μ_m, the log-likelihood function $L(\mathcal{X}|\Theta)$ must be derived after μ_m in the first step (McLachlan and Krishnan, 2007).

$$\frac{\partial lnP(\mathcal{X}|\Theta)}{\partial \mu_m} = \sum_{n=1}^{N} \frac{1}{\sum_{m'=1}^{M} \alpha'_m \mathcal{N}(x_n|\mu'_m, \Sigma'_m)} \frac{\partial \sum_{m=1}(x_n|\mu_m, \Sigma_m)}{\partial \mu_m}$$

$$\sum_{n=1}^{N} \underbrace{\frac{\alpha_m \mathcal{N}(x_n|\mu_m, \Sigma_m)}{\sum_{m'=1}^{M} \alpha'_m \mathcal{N}(x_n|\mu'_m, \Sigma'_m)}}_{r_m^n} \frac{\partial[ln(\alpha_m) + ln\mathcal{N}(x_n|\mu_m, \Sigma_m)]}{\partial \mu_m}$$

The probability $r_m^n = P(m|x_n, \Theta)$ is the posterior probability for component m given x_n and the parameters (McLachlan and Krishnan, 2007).

The derivation of the normal distribution are:

$$\frac{\partial \sum_{m=1}(x_n|\mu_m, \Sigma_m)}{\partial \mu_m} = \frac{1}{2}(\Sigma_m^{-1} + (\Sigma_m^{-1})^T)(x_m - \mu_m),$$

in which $(\Sigma_m^{-1} + (\Sigma_m^{-1})^T) = 2\Sigma_m^{-1}$, since Σ is symmetrical, that means $\Sigma = \Sigma^T$. It follows:

$$\frac{\partial \sum_{m=1}(x_n|\mu_m, \Sigma_m)}{\partial \mu_m} = \Sigma_m^{-1}(x_m - \mu_m)$$

Applying those 2 previous derivatives results:

$$\frac{\partial lnP(\mathcal{X}|\Theta)}{\partial \mu_m} = \sum_{n=1}^{N} r_m^n \Sigma_m^{-1}(x_m - \mu_m)$$

The derivation is set to 0 in the next step and multiplied both sides with Σ_m

$$\sum_{n=1}^{N} \Sigma_m^{-1}(r_m^n x_m - r_m^n \mu_m) \stackrel{!}{=} 0$$

After further reformulation, we get the formula to calculate μ_m

$$\mu_m \sum_{n=1}^{N} r_m^n = \sum_{n=1}^{N} r_m^n x_n$$

$$\mu_m = \frac{\sum_{n=1}^{N} r_m^n x_n}{\underbrace{\sum_{n=1}^{N} r_m^n}_{N_m}}$$

The mean value is calculated from the probability r_m^n weighted data x_n (Bishop, 2012). The N_m is the affective number of data points that are modeled by component m.

To calculate μ_m you need r_m^n, which in turn depends on Θ. This leads to the classic chicken-egg problem (Bishop, 2012). The consequence of this is that the calculation of μ_m is done iteratively, so that means Θ has to be initialize (McLachlan and Krishnan, 2007). This enables the iterative calculation of r_m^n and μ_m (Bishop, 2012).

Now the $L(\mathcal{X}|\Theta)$ is derived according to Σ and the derivative is then set to 0 (Bishop, 2006).

$$\frac{\partial \, ln \, P(\mathcal{X}|\Theta)}{\partial \, \Sigma_m} \stackrel{!}{=} 0$$

This derivation can be found in (Bishop, 2006). The solution is:

$$\Sigma_m = \frac{1}{N_m} \sum_{n=1}^{N} r_m^n (x_n - \mu_m)(x_n - \mu_m)^T$$

Required is again the posterior distribution to weight the data $\mathcal{X}$ (Bishop, 2012). For $M = 1$, that means for only 1 gaussian distribution would result a probability of $r_m^n = P(m|x_n, \Theta) = 1$, in which the probability can be calculated as follows:

$$\Sigma = \frac{1}{N} \sum_{n=1}^{N} (x_n - \mu)(x_n - \mu)^T$$

$$\mu = \frac{1}{N \sum_{n=1}^{N} x_n}$$

We obtain the Maximum-Likelihood solution of a gaussian distribution as being explained above (Dempster et al., 1977).

Furthermore, we would have to derive $L(\mathcal{X}|\Theta)$ according to α_m and then set it to 0.

$$\frac{\partial \, ln \, P(\mathcal{X}|\Theta)}{\partial \, \alpha_m} \stackrel{!}{=} 0$$

This is an optimization problem in α with the constraint $\sum_{m=1}^{M} \alpha_m = 1$. We can obtain the solution through the Lagrange Multiplicators. The Lagrange function is constructed by combining the log-likelihood function $L(\mathcal{X}|\Theta)$ with the constraint as follows (Buse, 1982):

$$J(m) = ln \, P(\mathcal{X}|\Theta) + \lambda \left(\sum_{m=1}^{M} \alpha_m - 1 \right)$$

The Lagrange function J_m consists of the log-likelihood function $L(\mathcal{X}|\Theta)$ plus a term consisting of the Lagrangian multiplicator λ and the condition $\sum_{m=1}^{M} \alpha_m - 1$ (Dempster et al., 1977). The function J_m is derived after α_m. First we calculate the derivative of $L(\mathcal{X}|\Theta)$ according to α_m.

$$\frac{\partial\, ln\, P(\mathcal{X}|\Theta)}{\partial\, \alpha_m} = \sum_{n=1}^{N} \frac{1}{\sum_{m=1}^{M} \alpha'_m \mathcal{N}(x_n|\mu'_m, \Sigma'_m} \frac{\partial\, \sum_{m=1}^{M} \alpha_m \mathcal{N}(x_n|\mu_m, \Sigma_m}{\partial \alpha_m}$$

$$= \sum_{n=1}^{N} \frac{\mathcal{N}(x_n|\mu_m, \Sigma_m)}{\sum_{m=1}^{M} \alpha'_m \mathcal{N}(x_n|\mu'_m, \Sigma'_m}$$

The derivative from J_m to α_m is obtained by the previous function and the derivation of the conditions$\sum_{m=1}^{M} \alpha_m - 1$ (Dempster et al., 1977). That means:

$$\frac{\partial\, J_m}{\partial\, \alpha_m} = \sum_{n=1}^{N} \frac{\mathcal{N}(x_n|\mu_m, \Sigma_m)}{\sum_{m=1}^{M} \alpha'_m \mathcal{N}(x_n|\mu'_m, \Sigma'_m} + \lambda$$

Now we set the derivative to 0

$$\frac{\partial J_m}{\partial \alpha_m} \stackrel{!}{=} 0$$

and multiply both sides of the equation with α_m. Thus we get the following formula:

$$\sum_{n=1}^{N} r_m^n + \lambda \alpha_m \stackrel{!}{=} 0$$

Then we sum up on both sides over m components and set $N_m = \sum_{n=1}^{N} r_m^n$. We get the following equation:

$$\sum_{n=1}^{N} N_m + \sum_{m=1}^{M} \lambda \alpha_m = 0$$

Since $\lambda \sum_{m=1}^{M} \alpha_m = \lambda$ under the condition $\sum_{m=1}^{M} \alpha_m = 1$ and $\sum_{m=1}^{M} N_m = N$, it follows that $\lambda = -N$. We set $\lambda = -N$ in the function and we get:

$$N_m - N\alpha_m = 0$$

$$\alpha_m = \frac{N_m}{N}$$

Similar to the derivation for μ_m and Σ_m, r_m^n is also necessary here, since r_m^n always adds all data weighted to the updates. This results in an iterative algorithm for

estimating Θ_{ML}. This algorithm is known as Expectation Maximization for GMMs (McLachlan and Krishnan, 2007) and will be introduced in the next chapter.

Expectation-Maximization (EM) Algorithm

The EM algorithm for learning GMMs is iterative. First the parameters theta are initialised. In the E-step, the parameter Theta r_m^n can be used to calculate (Minka, 1998). In the maximising step (M-Step) the parameters μ_m, Σ_m and α_m are recalculated with the help of r_m^n. The E and the M-Step are alternately performed until the log-likelihood function converges (Dempster et al., 1977).

The individual steps are briefly summarised below.

1. Initialisation: t.....iteration counter (McLachlan and Krishnan, 2007)

$$\Theta\{\alpha_m^{t=0}, \mu_m^{t=0}, \Sigma_m^{t=0}\}_{m=1}^{M}$$

2. E-Step: calculate class affiliation(McLachlan and Krishnan, 2007)

$$r_m^n = \frac{\alpha_m^t \mathcal{N}(x_n|\mu_m^t, \Sigma_m^t)}{\sum_{m'=1}^{M} \alpha'_m \mathcal{N}(x_n|\mu'_m, \Sigma'_m)} = P(m|x_n, \Theta^t)$$

   $P(m|x_n, \Theta^t)$ gives the probability for m given x_n and Θ. This posterior distribution is in fact equal to the posterior in the Bayes classifier with the assumption of a normal distribution as a likelihood model.

3. M-Step: calculation of the Parameter Θ (McLachlan and Krishnan, 2007)

$$\mu_m^{t+1} + \frac{1}{N_m} \sum_{n=1}^{N} r_m^n x_n$$

$$\Sigma_m^{t+1} = \frac{1}{N_m} r_m^n (x_n - \mu_m^{t+1})(x_n - \mu_m^{t+1})^T$$

$$\alpha_m^{t+1} + \frac{N_m}{N}$$

$$t = t + 1$$

4. Evaluation (Neal and Hinton, 1998):

$$L(\mathcal{X}|\Theta^t) = log(P(\mathcal{X}|\Theta^t))$$

$\rightarrow$ if converges, termination of $\Theta_{ML} = \Theta^t$

$\rightarrow$ if not $\Rightarrow$ E-Step

One possibility to initialise Θ^0 is: 1. α_m^0 in uniform distribution function $\alpha_m^0 = \frac{1}{M}$ (Neal and Hinton, 1998) 2. Σ_m^0 is set to the covariance matrix Σ of the data X, that means $\Sigma = \frac{1}{N}(x_n - \mu)(x_n - \mu)^T$, in which $\mu = \frac{1}{N}\sum_{n=1}^{N} x_n$ (R. Neal & G. Hinton,1998) 3. For μ_m^0 we can randomly select samples or use k-means Algorithm (Neal and Hinton, 1998)

K-means Algorithm

The goal of the K-means algorithm is to divide the data into clusters. The number of clusters is denoted by K. one way to derive the K-means algorithm is to modify the EM algorithm for GMMs (McLachlan and Krishnan, 2007). Under the following assumptions, the EM algorithm for GMMs becomes the K-means algorithm:

1. $\alpha_m = P(m) = \frac{1}{M} \ldots \forall m$; α_m is modelled by a uniform probability distribution and not modified,

i. a can be neglected (Hartigan and Wong, 1979)

2. $\Sigma_m = \sigma^2 I \ldots \forall m$; All components are represented by the same spherical covariance matrix (Hartigan and Wong, 1979)
3. Classifications of samples x_n to components m; that means $m = argmax_m, [r_m^n]$; each sample will be modelled from one component (Hartigan and Wong, 1979).

We modified the E-Step

$$r_m^n = \frac{\alpha_m \mathcal{N}(x_n|\mu_m, \Sigma_m)}{\sum_{m=1}^{M} \alpha_m \mathcal{N}(x_n|\mu_m, \Sigma_m)}$$

under the above conditions (Nasrabadi, 2007). Assumption 2 leads to the decision

function $g_m(x_n)$

$$g_m(x_n) = -\frac{(x_n - \mu_m)^T(x_n - \mu_m)}{2\sigma^2} + lnP(m)$$

By the first assumption P_m can be neglected. Furthermore, $2\sigma^2$ is just a scaling factor and can also be neglected(Nasrabadi, 2007). So we get the Euclidean distance as a decision-making function. The third assumption leads to classification of x_n to component m (Nasrabadi, 2007)

$$m^* = \underset{m}{argmax}[g_m(x_n)]$$

$$= \underset{m}{argmin}[(x_n - \mu_m)^T(x_n - \mu_m)]$$

The K-means algorithm for clustering data $\mathcal{X} = \{x_1, ..., x_N\}$ into K clusters is shown in this section. Here, Y_m represents the set of all data points x_n which are classified to the component m, that means $Y_m = \{x_n | m = argmin_m[(x_n - \mu_m)^T(x_n - \mu_m)]\}$. In the following the steps of the K-means algorithm are presented. In this case variable for component m in GMMs is replaced by the variable K (Nasrabadi, 2007).

1. Initialisation: $t \ldots$ iterationcounter Select K samples randomly for the cluster centres μ_k: $\Theta^0 = \{\mu_k^{t=0}\}_{k=2}^K, t = 0$ (Nasrabadi, 2007).

2. Classification of the Samples to the components (modified E-Step) (Nasrabadi, 2007)

$$Y_k = \{x_n | k = \underset{k'}{argmin}[(x_n - \mu_{k'}^t)^T(x_n - \mu_{k'}^t)]\} \ldots \forall k = 1, ..., K$$

3. Step 2: recalculation of the mean-vectors (gravity of the clusters) due to the allocation in Y_k (Nasrabadi, 2007)

$$\mu_k^{t+1} = \frac{1}{|Y_k|} \sum_{x_n \in Y_k} x_n$$

$$t = t + 1$$

4. Evaluation of the cumulative distance (Nasrabadi, 2007)

$$J^t = \sum_{k=1}^{K} \sum_{x_n \in Y_k} (x_n - \mu_{k'}^t)^T (x_n - \mu_{k'}^t)$$

→ if J^t converges, that means:$|J^t - J^{t_1}| < \epsilon$, so the optimal cluster centers $\{\mu_1^t, ..., \mu_K^t\}$

→ if no convergence, that means: $|J^t - J^{t_1}| > \epsilon, \Rightarrow$ Step 1

Spectral Clustering

Spectral clustering can be described as a group of partitioning, deterministic procedures. The term "Spectral" is based on the fact that the clustering is calculated from the spectrum of the similarity matrix. For the first time, the calculation of partitions using eigenvalues and vectors was proposed by Fiedler (1973) (Fiedler, 1975) and Donath and Hoffmann (1973) (Donath and Hoffman, 1972).

For spectral clustering, it is not only necessary to have a vector with measured values/data for the objects to be clustered, but a further step must already be taken: pairwise similarities (between 0 and 1) or distances have been calculated (Von Luxburg, 2007). This can be done in different ways. Above all that, it is important to ensure that relatively high similarity values also correspond to the desired cluster criteria. Inaccuracies with low similarities are less severe, because in most cases only high values have an impact on the result (Von Luxburg, 2007). How distances are calculated depends very much on the form in which the data is available, and whether the source of the data may imply a particular variant of the calculation of the partition (Von Luxburg, 2007).

1. Calculation of the similarity matrix (Von Luxburg, 2007) $A_{n \times n}$

$$A_{i,j} = exp^{\frac{-|x_i - x_j|^2}{2\sigma^2}}, if\ i \neq j\ and\ A_{ii} = 0$$

$$\sigma^2 : scaling\ factor$$

2. Calculation of diagonal matrix (Von Luxburg, 2007) $D_{n \times n}$

$$D_{i,j} = \sum_{l=1}^{n} A_{ij},\ if\ i = j,\ and\ A_{ii} = 0$$

that means, in the diagonal of D, the line sums of A.

Calculate $L = D^{-\frac{1}{2}}\ AD^{-\frac{1}{2}}$.

$$D^{-\frac{1}{2}} := \frac{1}{\sqrt{D_{ij}}},\ if\ i = j,\ and\ A_{ii} = 0$$

3. find $v_1, ..., v_k$, die k biggest eigenvectors of L, so that all v_i are pairwise diagonal (Von Luxburg, 2007). Create a matrix out of it

$$X_{n\times k} = [v_1, ..., v_k] \in \mathbb{R}^{n\times k}$$

4. Construct the Matrix $Y_{n\times k}$ via Normalisation of X (Von Luxburg, 2007):

$$Y_{ij} = \frac{X_{ij}}{\sqrt{\Sigma_j\ X_{ij}^2}}$$

5. every row Y_i of Y is a point in $\mathbb{R}^k$. Cluster the points with any simple cluster algorithms like $k - means$ (Von Luxburg, 2007).

6. Assign the cluster j to each original point x_i, if and only if the line Y_i lies in the cluster j (Von Luxburg, 2007).

Cluster Validation Metrics

In some situations before starting with the clustering tasks, it is necessary to determine if the data really show any tendency to be clustered. After performing a cluster analysis, it is important to validate the quality of cluster analysis results. Clustering methods will always find groups, even when there are no patterns. Clustering validation can be a first step to determine the correct number of clusters. It is also a tool to compare different clustering methods applied to the same dataset (Halkidi et al., 2001).

Cluster analysis consists of three phases: preparation, clustering and cluster validation (Halkidi et al., 2001). The preparation includes the selection of the relevant variables and the normalisation of the data. Subsequently, a clustering algorithm and its parameters have to be selected so that the clustering can be performed. Thereafter, the cluster validation takes place, because after performing a clustering, it is not known how well the clusters match the data. The cluster algorithms cannot ensure that the "perfect" partitioning is found. Therefore several algorithms and their parameters have to be tried out. The clustering algorithms used will always find a solution, even if there is no structure in the data. To uncover these faulty clusterings, cluster validation is necessary (Halkidi et al., 2001).

For this purpose, it is evaluated how well the determined cluster match the underlying data. Also calling the algorithm with an incorrect parameter, too high a cluster number, will lead to a sub-optimal solution (Halkidi et al., 2001). The reason is that most algorithms do not detect the "perfect" number of clusters. The algorithms must be compared to different cluster numbers. The solutions must be compared using cluster validation to determine which cluster number is optimal (Halkidi et al., 2001). A representation of the data to determine visually correct number of clusters is only possible up to three dimensions. The restriction is regularly exceeded in real applications, so indices for cluster validation are used (Halkidi et al., 2001).

The internal validation metrics focus on the information contained in the cluster and identify the issue of how data points are constructed based on that information. A good cluster analysis result is finding clusters where the data points within a cluster are close to each other (Liu et al., 2010).

The result of the internal cluster validation shows only the matching accuracy of the original cluster algorithm and the validation function. Therefore, the internal validation methods in this master thesis will be considered a reference only, but not the main assessment criteria (Liu et al., 2010).

The clustering indices was based on *clusterCrit* package from the Bioconductor framework (Desgraupes, 2013) converted into python. From all available indices in *clusterCrit*, only specific indices have been chosen for the validation that will be described in the following section later.

Additionally to the internal clustering indices which include its internal dataset quan-

tity vectors with the basis of statistical testing, the relative clustering indices are based on comparing a cluster to other clusters and thus, it does not include statistical testing. The following indices are being applied in this master thesis are a reformulation BIC (Bayesian Information Criterion) and AIC (Akaike Information Criterion) which is a decisive first local maximum knee point clustering detection proposed from Q. Zhao, V. Hautamaki and P. Fraenti (Zhao et al., 2008).

In external validation, the results of a cluster can be validated based on data what was not used for the clustering, usually such as labels of known classes or external markers. Such markers consist of labels classified by human experts and in our case the labels of other clustering algorithms. These types of evaluation methods measure how close the clustering of the set of predetermined classes is.

Clustering Validation metrics notations

X is labeled as the data matrix. The size of X are $N \times p$, where N are the observations and p are the features. X is assumed to be clustered in K groups. L_K are the cluster length (Krzanowski and Lai, 1988).

M is labeled as the center of the gravity of all clusters. $M_1, ..., M_N$ are the coefficients that represent all observations. G is the centroid or the barycentre of all points (Krzanowski and Lai, 1988).

Total Dispersion Matrix - Total Sum of Squares

$$TSS = Trace(T) = \sum_{i=1}^{N} ||M_i - G||^2$$

So TSS is defined as total scattering is the total sum of squares of the points around the centroid (Krzanowski and Lai, 1988).

Within Group Dispersion Matrix - Within Group Sum of Squares

$$WGSS^{\{k\}} = Trace(WG^{\{k\}}) = \sum_{i \in I_k} ||M_i^{\{k\}} - G\{k\}||^2$$

where I_k is defined as the set of indices of the observations from cluster of the sub-matrix of $X^{\{k\}}$, which is denoted as C_k (Krzanowski and Lai, 1988).

The within group sum of squares is calculated as the total of squared distances between $M_i^{\{k\}}$ and $G^{\{k\}}$ of the cluster (Krzanowski and Lai, 1988). So the total of the within group sum of squares is:

$$WGSS = \sum_{k=0}^{K} WGSS\{k\}$$

Between Group Dispersion Matrix - Between Group Sum of Squares

The between group matrix measures the cluster dispersion between groups, so the dispersion of the centroids $G^{\{k\}}$ of each cluster to the total set of data G (Krzanowski and Lai, 1988).

$$BGSS = Trace(BG) = \sum_{k=1}^{K} n_k ||G^{\{k\}} - G||^2$$

The between group sum of squares is the total weighted sum of squared distances between $G^{\{k\}}$ and G with the weight elements n_k in C_k clusters (Krzanowski and Lai, 1988).

Log-likelihood function

For calculating the knee point based clustering metrics BIC (Bayesian Information Criterion) and AIC (Akaike Information Criterion), the log-likelihood function hast to be defined by the following formula:

$$L(\Theta_i) = \sum_{i=1}^{n_i} logpr(x_i)$$

$$= \sum_{i=1}^{K_i} log\left(\frac{K_i}{N} \frac{1}{N(2\pi)^{\frac{d}{2}} \sum^{\frac{1}{2}}} exp\Big(-\frac{||M_i - C_{p(i)}||^2}{2\Sigma_i}\Big)\right)$$

$$= K_i \, log \, K_i - K_i \, log \, N - \frac{K_i * d}{2} log(2\pi) - \frac{K_i}{2} log \, \Sigma_i - \frac{K_i - L_k}{2}$$

where Σ is the maximum likelihood estimate for the variance of the i^{th} cluster:

$$\Sigma = \frac{1}{K_i - L_k} \Sigma_{j=1}^{K_i} ||x_j - C_i||^2$$

K_i is the size of each cluster, M_k is the j^{th} point in the cluster and C_i is the i^{th} cluster.

Clustering indices

This chapter describes the different clustering validation indices that for evaluating the optimal numbers of clusters (Liu et al., 2010).

Ball and Hall

The Ball and Hall index was introduced on the basis of the average distance of the points to the centroid (Desgraupes, 2013). It is computed as:

$$BH = \frac{1}{K} \sum_{k=1}^{K} \frac{1}{n_k} \sum_{i \in I_k} ||M_i^{\{k\}} - G^{\{k\}}||^2$$

Banfeld and Raftery

the Banfeld and Raftery index is on the basis of the weighted sum of the logarithms of the traces of within group dispersion of each cluster (Banfield and Raftery, 1993). It can be calculated as:

$$BR = \sum_{k=1}^{K} n_k \, log \left(\frac{Trace(WG^{\{k\}}}{n_k} \right)$$

Calinsky and Harabasz

Calinksy and Harabasz index was proposed as:

$$CH = \frac{N-K}{K-1}\frac{BGSS}{WGSS}$$

which K is the value, that maximises the index, the specify the optimal number of clusters (Caliński and Harabasz, 1974).

Friedman and Rubin - Det Ratio

The Det Ratio and log Det Ratio index was introduced by Friedman and Rubin on the basis of a non hierarchical clustering validation method (Friedman and Rubin, 1967).

$$FR = \frac{det(T)}{det(WG)}$$

$$logFR = N\ log\left(\frac{det(T)}{det(WG)}\right)$$

Hartigan - log SS Ratio

The log SS-Ratio (Milligan and Cooper, 1985) was introduced by Hartigan and can be computed as:

$$Hartigan = log\left(\frac{BGSS}{WGSS}\right)$$

Ratkowsky and Lance

The Ratkowsky and Lance (Ratkowsky and Lance, 1978) index was proposed as:

$$RL = \sqrt{\frac{\frac{1}{p}\sum_{j=1}^{p}\frac{BGSS_j}{WGSS_j}}{K}}$$

where:

$$BGSS_j = \sum_{k=1}^{K} n_k(\mu_j^{\{k\}} - \mu_j)^2$$

$$TSS_j = \sum_{i=1}^{N}(a_{ij} - \mu_{ij})^2$$

$BGSS_j$ stands for the between group sum of squares for each variable (Ratkowsky and Lance, 1978). The optimal value of K for the maximal value of the index (Milligan and Cooper, 1985). TSS_j stands for the total sum of squares for each variable (Milligan and Cooper, 1985).

Ray and Turi

The Ray and Turi index can be described as:

$$RT = \frac{1}{N}\frac{WGSS}{\underset{k<k'}{min}||G^{\{k\}} - G^{\{k\}'}||^2}$$

Within group sum of squares of all points divided by the number of the observations is the numerator and the minimum of squared distances between all the cluster centroids (Ray and Turi, 1999).

Scott

The Scott index is defined as the total of logarithm determinant of within group dispersion in each cluster (Desgraupes, 2013).

$$Scott = \sum_{k=1}^{K} log\, det\left(\frac{WG^{\{k\}}}{n_k}\right)$$

Trace W

the Trace W index can be defined as the within group sum of squares or the trace of within group dispersion (Milligan and Cooper, 1985).

$$TW = Tr(WG) = WGSS$$

AIC (Akaike Information Criterion)

There is a distribution of a variable with an unknown density function p in the basic population. In the maximum likelihood estimation (ML-estimation)(Pan and Fang, 2002) it starts from a known distribution with an unknown parameter Θ_u; so we assume that the density function can be written as $q(\Theta_u)$. The Kullback-Leibler divergence (Joyce, 2011) $D(P||Q)$ is used as distance measure between p and $q(\Theta)$. Θ is the estimated parameter from the maximum likelihood estimation. The better the model, the smaller the KL-divergence.

Akaike (Akaike, 2011) was able to show that the log-likelihood function $L(\Theta)$ is a distorted estimator for the KL-divergence and that the distortion is asymptotic converges to the number of parameters k to be estimated. Therefore, the AIC results with the logarithmic likelihood function as:

$$AIC = \sum_{i=1}^{L_k}(L(\Theta_i)) - penalty$$

where *penalty* is $(K-1)(K*d)$, and d is the dimension of the data set.

BIC (Bayesian Information Criterion)

The disadvantage of AIC is that the penalty is independent of the sample size (Bielza and Larrañaga, 2004). In the case of large samples, improvements of the log-likelihood function are possible, which is why the criterion for large samples tends to favour models with relatively many parameters (Bielza and Larrañaga, 2004).

BIC (Bayesian Information Criterion) also known as the Schwarz Information Criterion (SIC) can be defined as:

$$BIC = \sum_{i=1}^{L_k}(L(\Theta_i)) - \frac{1}{2} * penalty * \ logN$$

where the penalty factor increases logarithmically with the number of observations N.

Fowlkes and Mallows

The Fowlked-Mallows index computes the similarity between the groups returned by the clustering analysis results. The higher the value of the Fowlkes-Mallows index the more similar are the groups. It can be calculated using the following formula:

$$FM = \sqrt{\frac{TP}{TP + FP} * \frac{TP}{TP + FN}}$$

Where TP is the number of True Positives, FP is the number of False Positives, and FN is the number of False Negatives. The Fowlkes-Mallows index is the geometric mean of the precision and recall P and R, while the F-measure is its harmonic mean.

PCA

Principal component analysis (PCA) is a variable-oriented method that attempts to extract few latent factors in variables with many properties. For this purpose, main components are formed in descending order, that means that the first major component accounts for most of the variations (Wold et al., 1987).

Mathematically speaking, this means that the correlation of multidimensional features is minimised by conversion into a new-basis vector space. From the eigenvectors of the covariance matrix, a new matrix can be formed, which indicates the main axis transformation. This matrix must be recalculated for each record, making the principal component analysis problem-dependent (Abdi and Williams, 2010).

Suppose there are m data - i.e. a point cloud with m points - given in a P-dimensional space. To form the main components, the following procedure is used:

1. First, the origin of the coordinate system is placed in the center of gravity of the point cloud. Next, the coordinate system is rotated so that the first coordinate points in the direction of the greatest variance of the point cloud.

The first coordinate thus represents the first main axis, the variance the first main component (Abdi and Williams, 2010).

2. For the second main component, the coordinate system is rotated further so that now shows the second major axis in the direction of the remaining maximum variance. Thus, the second major axis and the second major component are fixed. This process is repeated until a new base is created (Abdi and Williams, 2010).

In terms of mathematics, this means the following: The data $x_i \in \mathbb{R}^N, i = 1, ., , , m$, correspond to the m points of the point cloud. These data are centered, that means that $\sum_{i=1}^{m} x_i = 0$. PCA finds the major axes by diagonalising the covariance matrix (Richardson, 2009),

$$C = \frac{1}{m} \sum_{j=1}^{m} x_i x_j^T$$

This can be diagonalised with nonnegative eigenvalues λ, since it is positive definite. The eigenvalues are determined by taking the equation for the eigenvalues $\lambda \geq 0$ and the eigenvectors $v \in \mathbb{R}^N \setminus \{0\}$ (Richardson, 2009)

$$\lambda_v = C_v$$

is calculated. By substituting equation the first into the second (Richardson, 2009), we obtain:

$$\lambda_v = C_v = \frac{1}{m} \sum_{j=1}^{m} \langle x_j, v \rangle x_j$$

Thus, all solutions v with $\lambda \neq 0$ are in the range of $x_1, ..., x_m$ (Richardson, 2009). consequently the second equation is equivalent to:

$$\lambda \langle x_i, v \rangle = \langle x_i, C_v \rangle$$

for all $i = 1, ..., m$.

t-SNE

t-SNE (t-Distributed Stochastic Neighbour Embedding) is a tool for visualising high-dimensional data. It converts affinities of data points into probabilities. The affinities in the original space are represented by common Gaussian distributions and the affinities in the embedded space by student t distributions (Maaten and Hinton, 2008). Therefore, t-SNE can take the local structure well into account and has even more advantages compared to the current methods:

1. Visualise the structure with different scales in a single card (Maaten and Hinton, 2008)
2. Visualising data that resides in several different manifolds or clusters (Maaten and Hinton, 2008)
3. Low tendency to collect points in the middle (Maaten and Hinton, 2008)

Given a high-dimensional data set $X = \{x_1, x_2, ..., x_N\}$, where $x_i \in R_p$ and the entries of the load collective of i. Contain vehicle. The set of corresponding low-dimensional data representations to be determined is denoted by $Y = \{y_1, y_2, ..., y_N\}$, where $y_i \in R_m$ and $m \ll p$ apply. To make it easy to visualise Y, 2 is typically chosen to be 2 or 3 (Maaten and Hinton, 2008).

The basic idea of t-SNE is to model "similarities" between any two objects x_i and x_j of the high-dimensional output data set X or between the sought-after low-dimensional representations y_i and y_j such that they each form a probability distribution over the object pairs (Maaten and Hinton, 2008).

The latter are defined as assigning a high probability to two "similar" and "adjacent" instances, whereas far from one another, i.e., in the case of the "similar" or "neighbouring" instances. very "dissimilar" data objects have a low probability under this distribution (Maaten and Hinton, 2008). Formally, the common probability p_{ij}, which uses t-SNE as a measure of the pairwise similarity between two high-dimensional objects x_i and x_j, is given by:

$$P_{ij} = \frac{P_{j|i} + P_{i|j}}{2N}$$

where $p_{i|j} = 0$ and the conditional probability $p_{j|i}$ by the normalised Gaussian kernel

(Maaten and Hinton, 2008).

$$Pj|i = \frac{exp\Big(- d(x_i, x_j)^2 / 2\sigma_i^2 \Big)}{\sum_{k \neq i} exp(-d(x_i, x_k)^2 / 2\sigma_i^2)}$$

is defined. Where $d(x_i, x_j)$ denotes a distance function, e.g. the Euclidean distance $d(x_i, x_j) = ||x_i - x_j||^2$, and the bandwidth of the Gaussian kernel is given by σ_i (Maaten and Hinton, 2008). The latter is chosen individually for each object i such that the perplexity of the conditional probability distribution P_i corresponds to a predefined value u. As a result, σ_i tends to have lower values for objects located in denser regions of the high-dimensional space than for objects located in sparsely populated areas. In that sense, perplexity can be seen as a measure of the effective number of neighbours of an object. It is by the equation:

$$Perp(P_i) = 2^{-\sum_j p_{j|i} log_2 p_{j|i}}$$

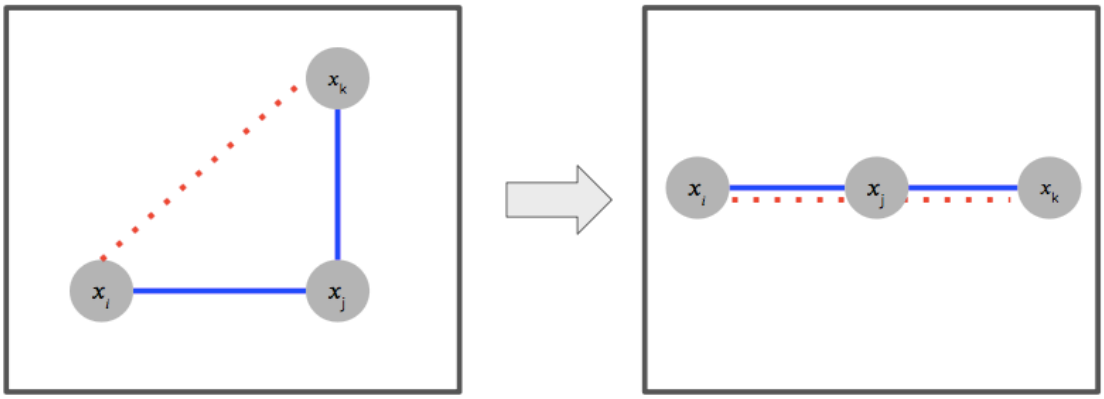

Figure 2.11: Sketch of distance problem in a projection of three two-dimensional points into the one-dimensional space: An exact modelling of short distances (line) in one-dimensional space leads to an increase in distance between more distant points (dashed-line)

The objects x_i and x_j as well as x_j and x_k are equidistant from each other in the two-dimensional space (see left diagram), while the distance between x_i and x_k is somewhat larger. If one wants to obtain the short distances (line) in one-dimensional space, the points x_i and x_k must be further farther than originally modelled from each other, since one dimension is not sufficient to obtain this distance as well. If,

on the other hand, one were to preserve the distance (dashed-line) between these two points in one-dimensional space, then all three points would be "closer together". This can lead to a complete overlapping or overlapping of similar points in the low-dimensional space, so that local differences or structures are no longer recognisable. This is also referred to as the so-called "crowding problem" (Maaten and Hinton, 2008).

In t-SNE, the similarities between the low-dimensional representations y_i and y_j of two objects x_i and x_j are therefore calculated by a normalised kernel of a student t-distribution with one degree of freedom (Maaten and Hinton, 2008):

$$q_{ij} = \frac{\left(1 + ||y_i - y_j||^2\right)}{\sum_{k \neq l} \left(1 + ||y_k - y_l||^2\right)^{-1'}}$$

where $q_{ii} = 0$. Since there is more mass on the flanks compared to the normal distribution in this probability distribution, unlike the original space, dissimilar objects can be modelled farther apart, counteracting the "crowding problem" (Maaten and Hinton, 2008).

Finally, we obtain the final coordinates of the projection points $y_1, y_2, ..., y_N$ in the low-dimensional space by minimising the Kullback-Leibler divergence (KL divergence) between the induced common probability distributions P and Q (Maaten and Hinton, 2008):

$$\min_Q(P||Q) = \sum_i \sum_{j \neq 1} p_{ij} log \frac{P_{ij}}{q_{ij}}$$

This minimisation problem can be solved by using a gradient descent method (Maaten and Hinton, 2008).

Label switching problem

After the optimal number of clusters of each experiments has been selected via Clustering Validation, the following task is to find out a way of aggregating those clustering results from all experiments into one unified clustering. The purpose of this

chapter is the determination of a general problem that occurs to most of the clustering algorithms, especially in mixture models and to introduce a solution of this problem.

The observed data in a mixture model are considered to derived from a heterogeneous population with mixing density functions on probabilities.(Jasra et al., 2005) Standard maximum likelihood techniques are being used to estimate the parameters for these models. The problem is that the estimated parameters can't be classified if more than one choice of the parameters acquired the same likelihood function. That means that all finite mixture models are non-classifiable if the component labels are permuted under symmetric priors. These permutations are also known as label switching (Jasra et al., 2005).

The label switching problem occurs, since mixture model components can be randomly ordered. While iterating the clustering algorithms, the label order with switch several times in each iterations. To acquire reasonable components, different methods have been proposed in solving these label switching problems. One method that is being implemented in this thesis has been called the relabelling algorithm (Jasra et al., 2005).

Relabelling algorithm

One of the first relabelling algorithms was developed by Stephens. The idea behind the algorithm is based upon the agreement on the $n \times k$ matrix of classification probabilities. Stephens applied the KL-divergence to measure the loss of the classification probabilities when the true probabilities are $P(\theta)$ (Stephens, 2000).

m starting points has to be chosen as the algorithm only assembles to a local maximum. The permutations and the quantities are being selected for the optimal result (Stephens, 2000).

1. Select m permutations $\tau^{(t)}\ t = 1, ..., m$
2. For $t = 1, ..., m,\ k = 1, ..., K$, calculate $q_{ik} = \frac{1}{m} \sum_{t=1}^{m}\ p_{i\tau_k}^{(t)}$
3. For $t = 1, ..., m$ find a permutation $\tau^{(t)} \in T_K$ that minimizes $\sum_{i=1}^{n} \sum_{k=1}^{K} p_{i\tau_k}^{(t)} log\left(\frac{p_{i\tau_k}^{(t)}}{q_{ik}}\right)$
4. If there is a progress to $\sum_{t=1}^{m} \sum_{i=1}^{n} \sum_{k=1}^{K} p_{i\tau_k}^{(t)} log\left(\frac{p_{i\tau_k}^{(t)}}{q_{ik}}\right)$ go to step 2, finish oth-

erwise (Papastamoulis, 2015).

The relabelling algorithm by Stephens (Stephens, 2000) refers to an automative application of an identifiability constraint, so that the permutations of the samples have equal labelling. It can be relatively compared with that of a k-Means clustering algorithm.

So the main application of the relabelling algorithm in the thesis is by defining one experiment as a cluster template and based on this template, all other experiments has to be relabelled at the same label as the cluster template. For the relabelling algorithm, it requires to have a probability estimation of each sample corresponding to the cluster labels. For all clustering algorithms that have been implemented into our experiments, only *k-means* and *Spectral-Clustering* algorithm does not provide a probability estimation function in *sklearn*. the *GaussianProcessClassifier()* from *sklearn* has been applied to calculate the probability estimation from the clustering results of all clustering algorithms. It is based on Laplace Approximation (Azevedo-Filho and Shachter, 1994) that applies a Gaussian approximation to the posterior over the latent variables to generate a probabilistic prediction, with a second order Taylor expansion around posterior maximum to obtain the probability estimation.

Gene Ontology (GO)

Now after the clustering analysis we left out with a question if the genes are significantly involved in any specific process. And if it is what have been expected, that is, in this case, if the genes will apply any relation to the clustering results. Gene ontology is generate in a hierarchical way. The gene ontology (GO) was developed to facilitate the annotation, the incorporation of information to genes in a system way. It provides a controlled vocabulary that describes the gene and its attributes of the gene product in any organism. Gene ontologies consist of structured and controlled vocabulary of terms that describe gene products according to their:

- Biological processes: A series of molecular events with beginning and end.
- Cellular components: Location in cellular or macromolecular structures.
- Molecular functions: Molecular activities such as enzymatic activities.

Gene Set Enrichment Analysis (GSEA)

The GSEA is a statistical method for so-called weighted gene expression for pairwise comparisons of two conditions. These gene groups can be chosen arbitrarily, so they can be compiled manually or come from databases. E.g., in order to be able to compare two conditions, the gene groups have similar functions or equal regulatory mechanisms.

In this thesis, the obtained gene expression data sets were compared in order to identify possible correlations between the datasets. The goal of GSEA as an effective method for gene expression analysis is to demonstrate a preferential distribution of given sets of genes in the examined data sets. For this purpose, we used the package *topGO* to conduct GSEA (Alexa and Rahnenfuhrer, 2010). The package was given a ranking of the expressed genes, which was calculated from the corrected p-value. Subsequently, an enrichment score (ES) was calculated, which is based on a weighted statistic for the degree of enrichment of the gene set, using the *new()* function.

A gene set is considered to be enriched if many genes from this gene group have high ranks in the hierarchically sorted gene list. The maximum of the cumulative sum, indicates how high the enrichment of the examined gene group is in comparison to the whole gene set. The more genes in the groups occupy high rankes in the gene rankings, the higher is the maximum of the cumulative sum. However, not only genes with high ranks are interesting, but also ranks with low ranks are of interest, considering, e.g. up- and down-regulated genes. The up-regulated genes occupy high ranks and down-regulated genes occupy low ranks in the gene list. The GSEA is a running sum statistical based on a bootstrapping process of the gene groups.

In this case, after the calculation of the cumulative sum, permutation tests are performed by gene randomisations in order to determine whether the calculated value of the running sum actually has a significant height. This permeation test used in the ranks distribution corresponds to the Kolmogorov-Smirnov test. # Results and Discussions

This chapter presents the results of the analysis. The results are divided into eight sections. First, the results of the quality assessments are shown, which tells about how well the data has been preprocessed. The second part of the results is the Pearson's correlation coefficient calculation and will decide whether which data can

be used for the analysis. The third section is the performance of the meta-analysis on the calculated p-values of both data. The fourth and the fifth part deals with cluster validation and the visualisation of the clustering results. Here, the section is also subdivided into the results of the toy data set and the gene expression data sets. The sixth part concerns the determination of the cluster probabilities with the respective label switching method around the cluster labels in correct labelling to the other clustering results. The penultimate and the final part of the results are corresponding to the evaluation and the biological interpretation of the clustering results by implementing the survival analysis with a Kaplan-Meier analysis and a gene set enrichment analysis.

Quality assessment

This section is the quality assessment, which are used to find the poor quality gene expression data points. If there are gene-expression data that are of insufficient quality, they are removed for the analysis. The following figures below show heat map, density, box and MA plots representing signal intensity distributions summarisation of the data.

The figure 2.12 shows a heatmap that is created across 40 randomly selected samples. Similar data that can be classified into certain groups are being searched, without previously specifying fixed groups. The heatmap determines the Euclidean distance of the intensities between 2 arrays.

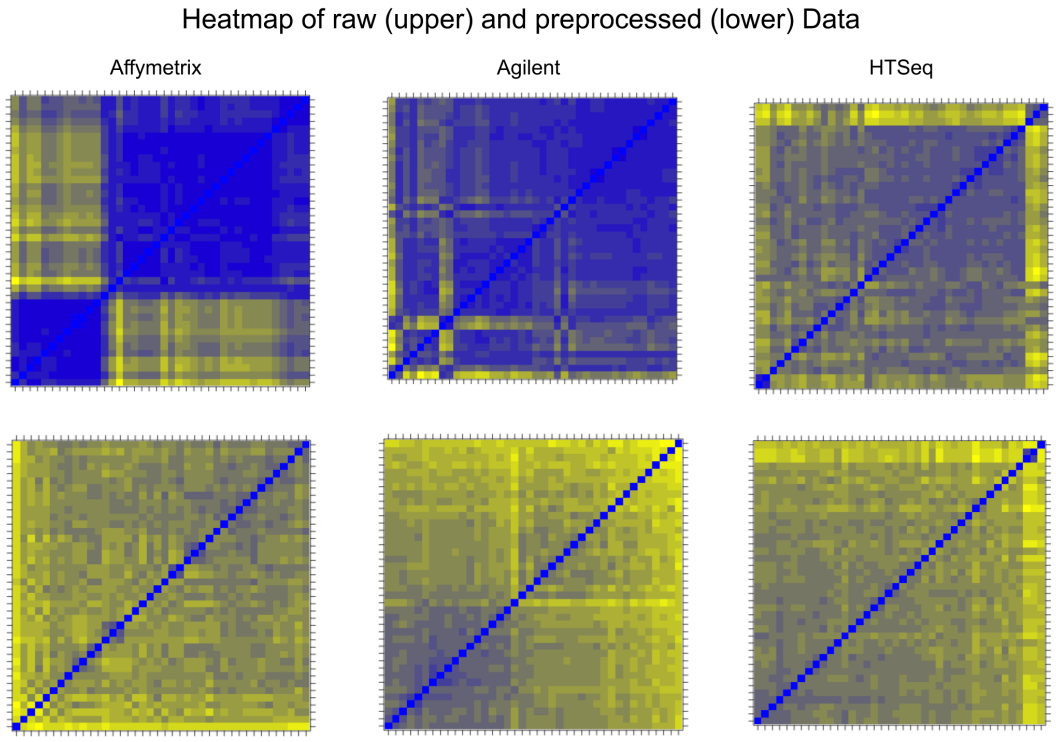

Figure 2.12: Heatmap of false colour of the distances between arrays. It shows the comparison between unprocessed (upper) vs. preprocessed arrays (lower) of a) Affymetrix, b) Agilent and c) HTSeq data.

The heatmap displays a false colour of the distances between arrays. The colour range (between blue - low correlation and yellow - high correlation) is being chosen to cover the distances encountered in the data. The rectangular patterns of the heatmap can be an indication of correlations for certain groups of samples or unintended experimental factors such as batch effects. The figure 2.12 is a comparison between unprocessed and preprocessed all experiments. The distance between the red and green channel is being calculated as the mean absolute difference between both arrays in Agilent microarray. Outlier detection of the heatmap is based on both channels which the sum of the distances to all other arrays are exceptionally large. 23 samples without the references were detected as outliers. The high correlation part on the edge of the raw data heat map of HTSeq could indicate the reference samples.

An MA plot allows to display the relationship between intensity and difference between two arrays, for e.g. the red and green channel of Agilent. It is a 2D plot with

a point for each genes. The x-axis is the average value across the channels or the log of mean of expression values (A = 1/2 * (log2(I1) + log2(I2))) and the y-axis the difference or the log fold change (M = log2(I1) - log2(I2)) between them. Genes with similar expression values will scattered around M = 0 value, meaning that genes expressed with no significant differences between the channels. Data values away from M = 0 line implies that genes are significantly expressed.

The figure 2.13 displays 2 MA-plots of (raw vs. preprocessed) Affymetrix. Each plot display the upper four worst and the lower four best MA-plots. The four worst plots show curved trends away from M = 0. After normalisation, we can observed that the upper worst plots have been normalised, which transform the expression values around M = 0.

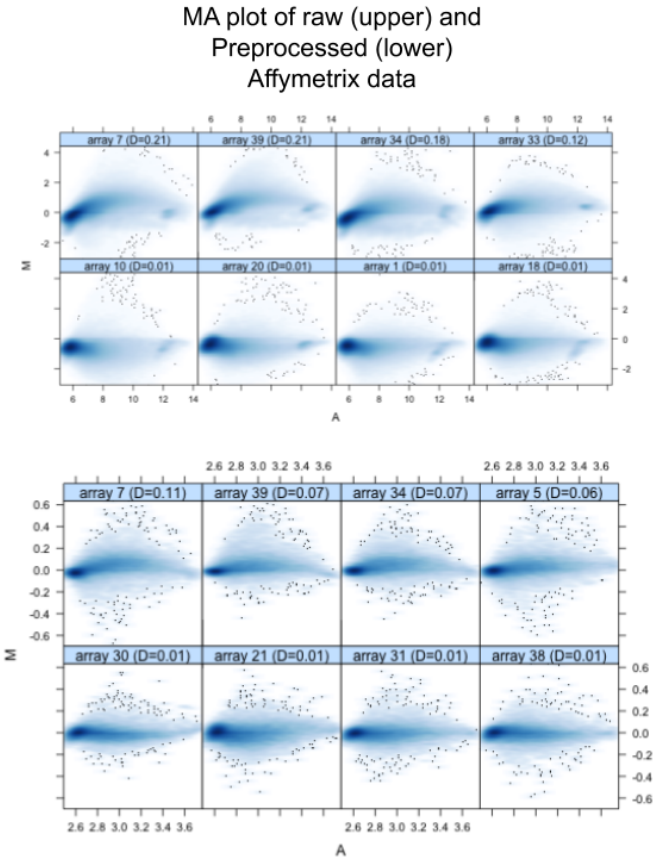

Figure 2.13: MA-Plot of Affymetrix. The upper plot shows the raw data und the lower plot the preprocessed data.

The MA-plot of Agilent on the figure 2.14 display more scattered MA-plots than Affymetrix. We can also observed a successful normalisation between the data, as the MA-plots of preprocessed data show linearity across M = 0.

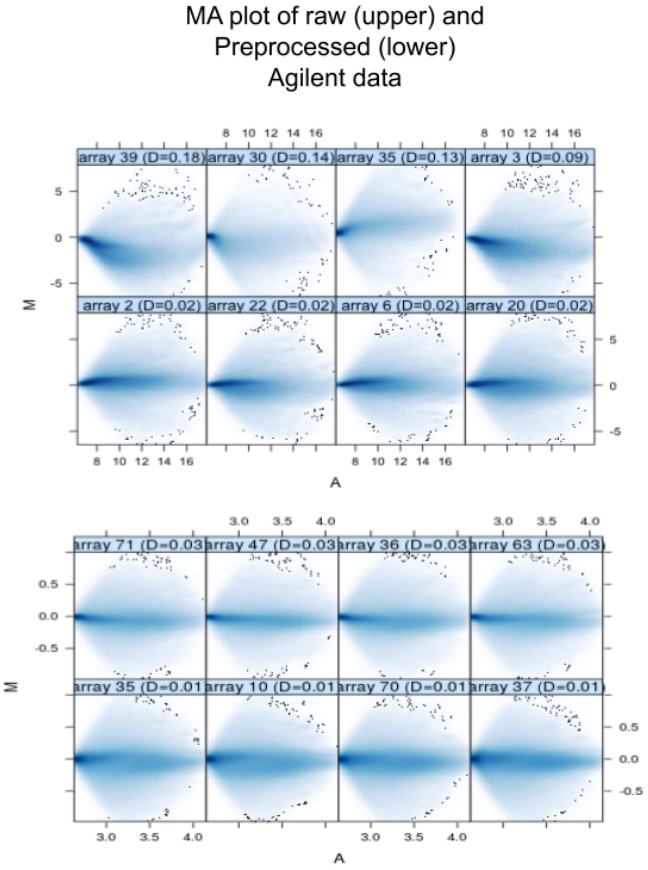

Figure 2.14: MA-Plot of Agilent. The upper plot shows the raw data und the lower plot the preprocessed data.

Similar to both figures 2.13 and 2.14, the HTSeq data on the figure 2.15 has been also normalised successfully, since it display more or less linearised MA-Plots.

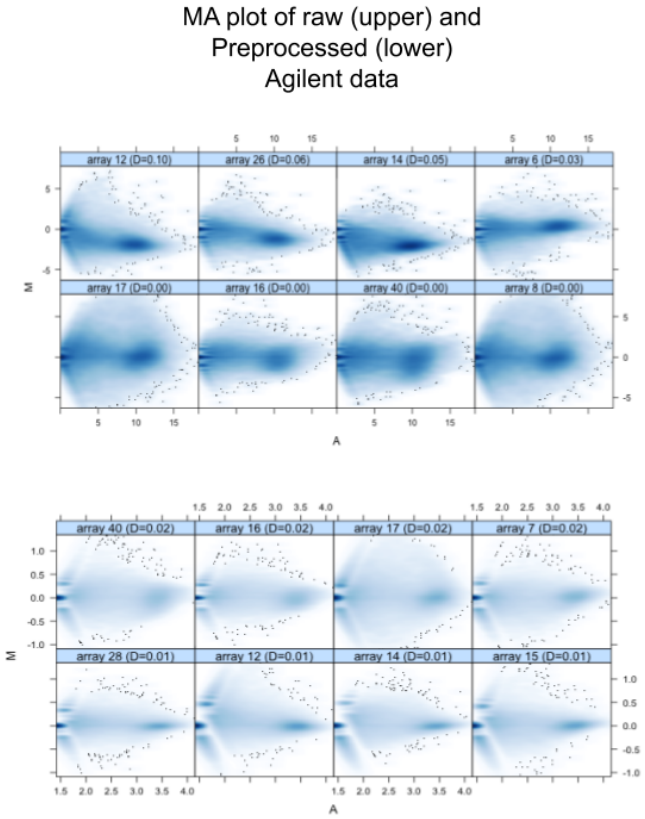

Figure 2.15: MA-Plot of HTSeq. The upper plot shows the raw data und the lower plot the preprocessed data.

Correlations

In order to implement the meta-analysis to combine multiple p-values across gene expression data, it is necessary that positive correlations between those data exist. The lower the correlation between the data, the lesser the method to generate appropriate results of the cluster analysis. The figures 2.16, 2.17, 2.18, 2.19 and 2.20 display the scatter plot between each paired samples of Affymetrix, Agilent, FPKM and HTSeq data against each other. What we can clearly observed is that Affymetrix and HTSeq show linear correlations between all random selected samples and genes, while the lowest correlations are displayed between FPKM and both Agilent and Affymetrix. Agilent against both HTSeq and Affymetrix display L-shaped scatter

plots. Many of the higher expressed values from Agilent show low expressions to both HTSeq and Affymetrix.

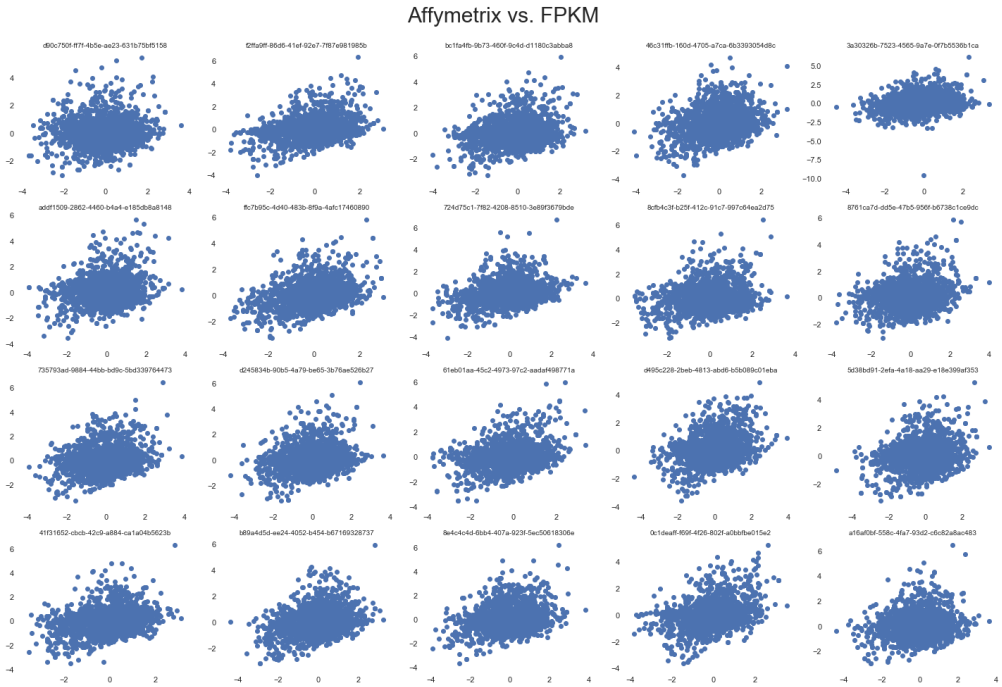

Figure 2.16: Scatter plot between Affymetrix and FPKM across each paired samples.

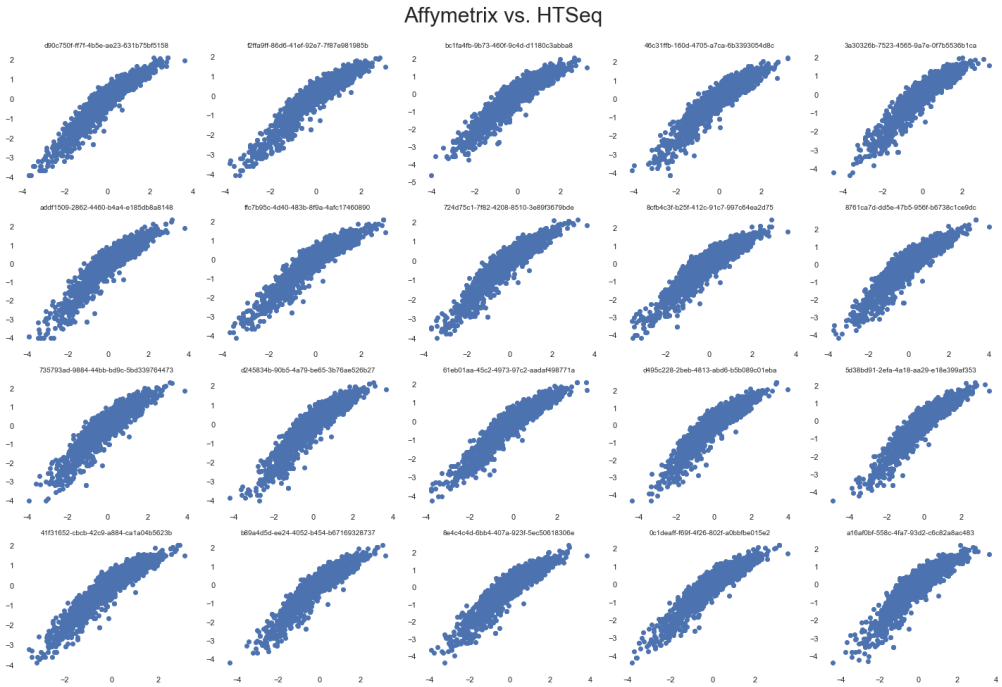

Figure 2.17: Scatter plot between Affymetrix and HTSeq across each paired samples.

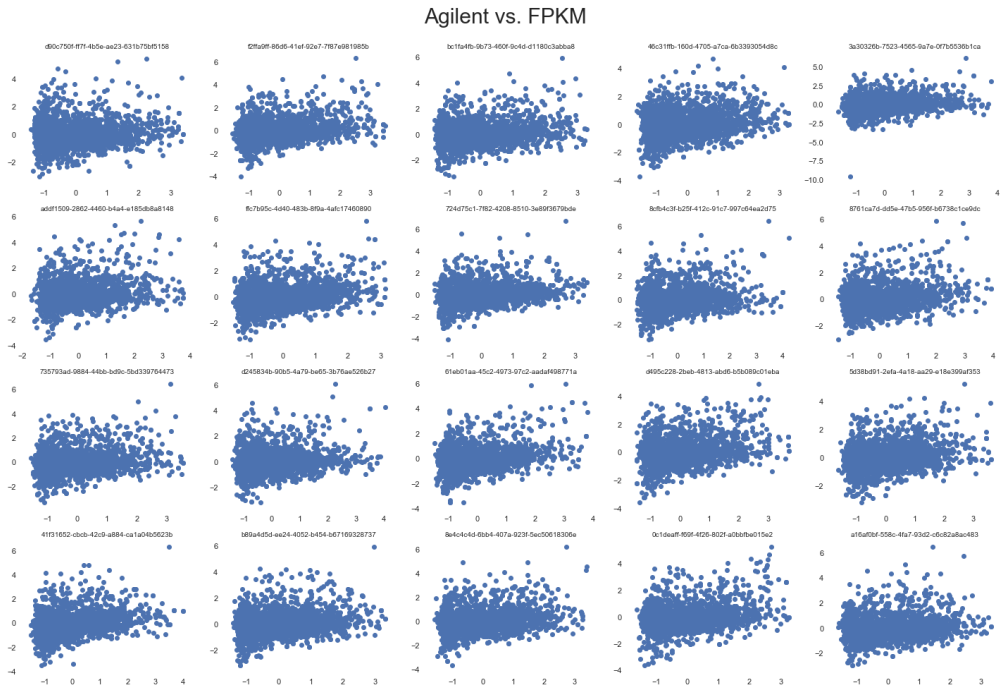

Figure 2.18: Scatter plot between Agilent and FPKM across each paired samples.

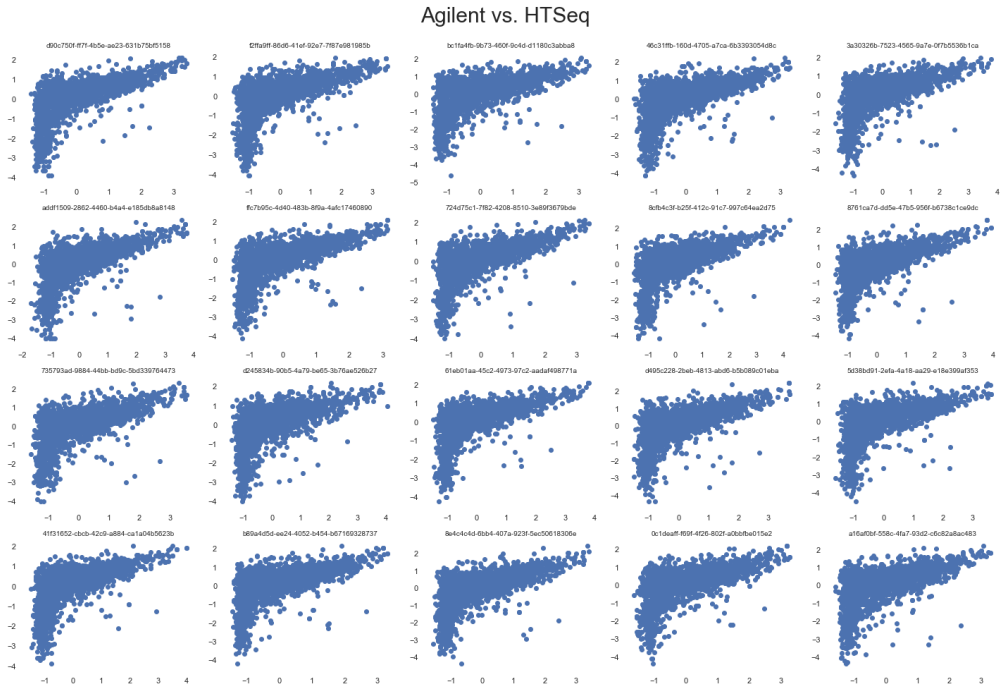

Figure 2.19: Scatter plot between Agilent and HTSeq across each paired samples.

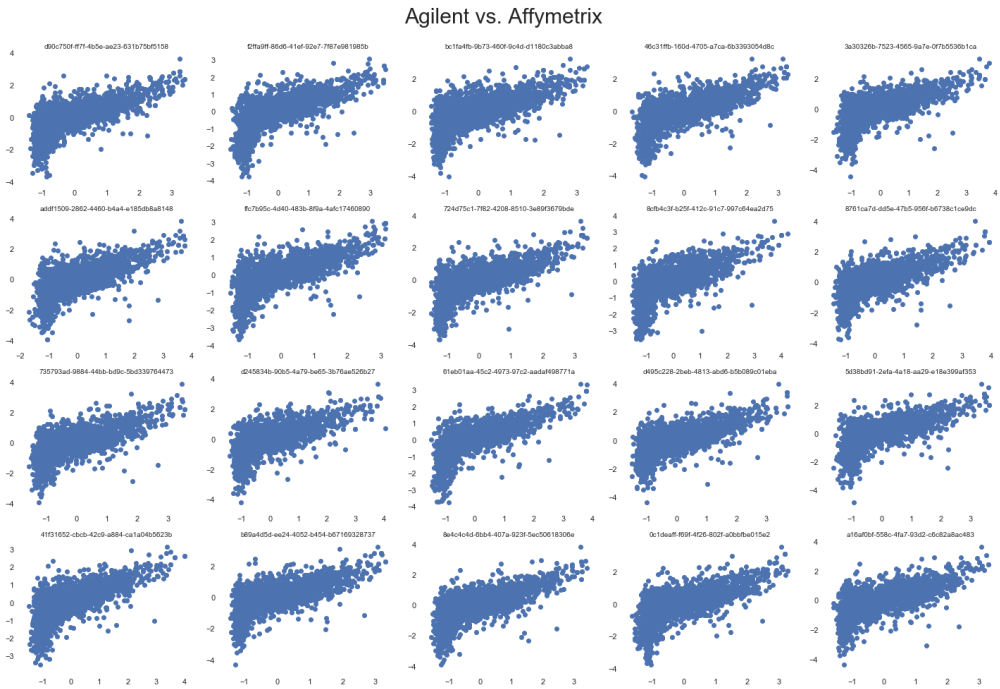

Figure 2.20: Scatter plot between Agilent and Affymetrix across each paired samples.

Furthermore, we calculated the correlations between each matched paired samples between the experiments using Pearson Correlation Coefficient. The results then underwent a Fisher's z-transformation for normally distributed coefficients and then visualise the results using empirical CDF, which show in the figure 2.21.

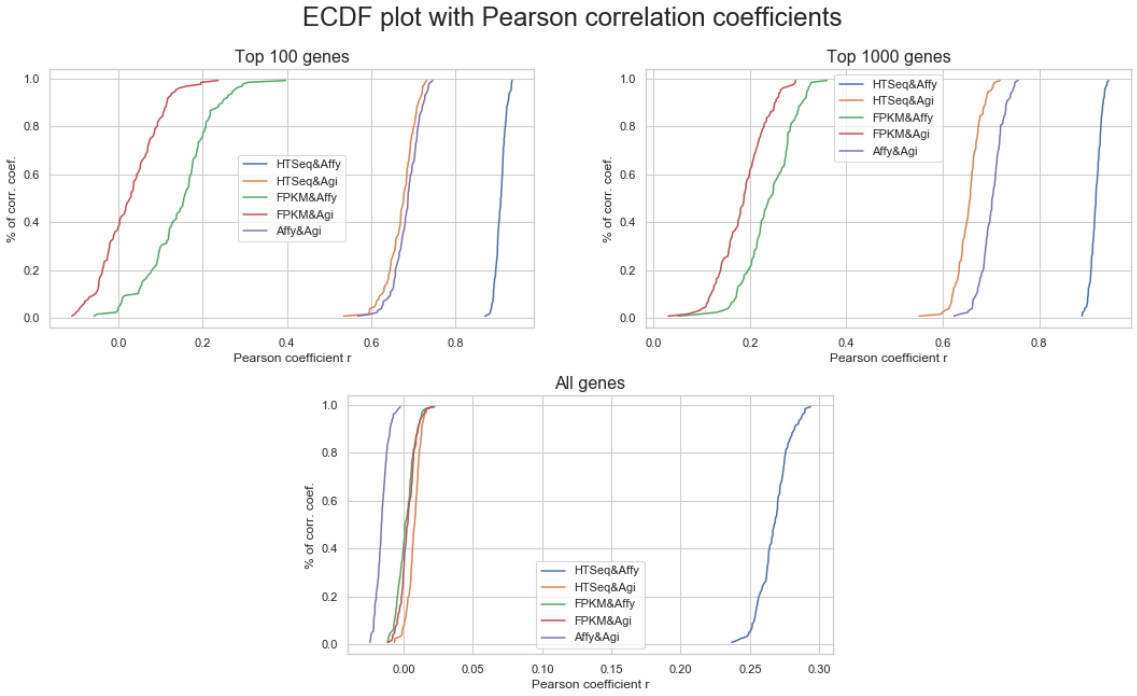

Figure 2.21: The plot shows 3 different kinds of plots. It shows the correlations of the experiments to each other. From 3 different gene expression data, there are 3 different ECDFs. The upper left plot shows the correlation of HTSeq against Affymetrix (blue), HTSeq against Agilent (orange), FPKM against Affymetrix (green), FPKM against Agilent (red) and Affymetrix against Agilent (purple) from the first 100 genes (left), the top 1000 genes (middle), all 7180 cross annotated genes (right).

Based on the ECDF plots shown from the figure 2.21, it display overall high Pearson coefficients between Affymetrix and HTSeq data, while lower Pearson coefficients could be observed between Agilent against Affymetrix and HTSeq. Worst values can be shown between FPKM against both Agilent and Affymetrix of the top 100 and 1000 genes. Slightly higher but clearly visible are the coefficients from all genes, in which HTSeq against Affymetrix stood out the most. Similar results have been approved from the publication of Y. Guo et. al., which stated that the low correlation between Agilent and HTSeq data is due to the normalisation difference between ratio and non-ratio representations of the data. Thus, HTSeq data are directly counted from the transcript abundance, while the gene intensity of Agilent microarray is based on the ratio between red and green channels.

Meta-Analysis for combining *p*-values

After Affymetrix and HTSeq (henceforth called RNA-Seq) were selected for further analysis, we determined the t-test statistics of the respective genes in which we used the reference samples/healthy samples against the cancer samples from both data to calculate the p-values to test the significance between healthy and tumour genes. After the p-values has been obtained, the p-values were then combined via meta-analysis annotated by its corresponding genes. This gave Affymetrix and RNA-Seq the same p-values of the genes, which can then be ranked and then used together to carry out the cluster analysis only with the top ranked genes.

Cluster Analysis and Validation

Based on the annotated RNA-Seq and Affymetrix data, the clustering analysis was performed on 127 samples and the top 100 most significant genes from the meta-analysis. The clustering analysis were simultaneously conducted using k-Means, Spectral Clustering and Gaussian Mixture Model with the their corresponding parameters. The k-Means was being run with a randomised initialisation, where k observations from data for initial cluster center were chosen at random, with 2000 number of iterations on a single run. Spectral Clustering was being run with the eigenvalue decomposition strategy using the *ARPACK* method. ARPACK is based on Fortran, that provides efficient eigenvalue decomposition of large sparse matrices. The affinity parameter determines the similarity between points in the matrix, and for our analysis, we used k-nearest neighbour. For the Gaussian Mixture Model we used both random and k-Means initialisation, with a 2000 iterations within a single run and a diagonal covariance matrix as the covariance type.

To find out the optimal cluster numbers, different clustering validation metrics have been introduced within this master thesis. Since each validation metric is dependent on the data size and structure, it is essential to evaluate the metrics to exclude those, which does not fit to our clustering validation. Additionally, since each clustering metrics has its own scoring results that differs mostly from each other in different scales. But the purpose for the clustering validation by comparing the metrics and visualising them in one figure, so a solution had to be found in order to solve this

problem. We implemented the *StandardScaler*() function from *sklearn*, since it assumes that the metrics are all normally distributed within each other and will scale them that the distribution is centred around 0, with a standard deviation of 1. The calculation of the mean and standard deviation is based on:

$$\frac{x_i - mean(x)}{stdev(x)}$$

The idea behind *StandardScaler*() is that it will transform the data in such, that the distribution will have a mean value of 0 and standard deviation of 1. Given the distribution of the data, each value in the dataset will have the sample mean value subtracted, and then decided by the standard deviation of the whole dataset. Requirements of the data transformation is that all metrics scores have to be arranged in a data frame column-wise.

Since the scores for the clustering validation are less important than the local maximum or minimum or the knee point of the metrics in our cases, the StandardScaler is acceptable just for the visual validation.

Results from Toy Dataset

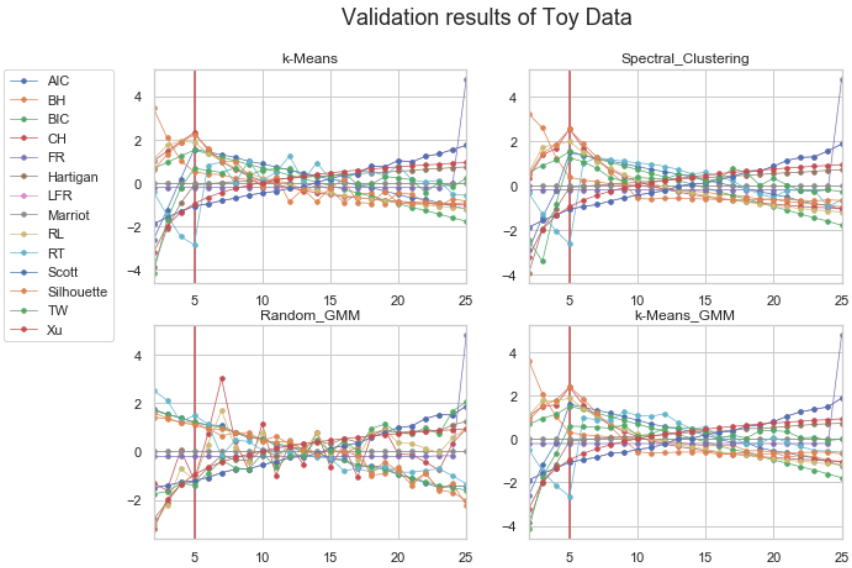

Figure 2.22: Clustering validation for all implemented metrics: Ball and Hall (BH), Banfeld and Raftery (BR), Calinski and Harabasz (CH), Hartigan, Friedman and Rubin (FR), Log Set Ratio (Log Friedman and Rubin - LFR), Ratkowsky and Lance (LR), Ray and Turi (RT), Trace_W (TW), Marriot, Xu, Silhouette, AIC and BIC.

The figure 2.22 represents the visualisations of the Toy Dataset being run through the clustering algorithms and being evaluated by the clustering metrics. The left graph represents all the clustering metrics introduced for the validation. The red shading serves as indication for a possible cluster number. The reason of creating the toy dataset with the same dimension as the gene expression data is to exclude those metrics that could not fulfil the clustering validation criteria. Based on the left graph, 7 out of 15 metrics do not perform as expected. Random GMM also show poor validation performance from the graph, as it could not identify the right number of clusters. We excluded those validation metrics, which seem to either demonstrate bad or only slightly acceptable validation performance, that could lead to misinterpretations of the clustering validation. Only 7 (BG, CH, Hartigan, RL, TW, AIC, BIC) validation metrics which could satisfy the expectations of a good validation performance which can be seen in the figure right graph.

Ball and Hall is the only metric that show an elbow-point scoring method, Hartigan is as well the only metric that show a knee-point based scoring method, while the

rest (AIC, BIC, Silhouette and Calinski and Harabasz scores) have a local maximum to determine the number of clusters. Given those validation agreements between the clustering validation metrics with the toy data set, we would seek to have similar results with both gene expression data.

Cluster allocation probabilities and Label Switching

We aggregated different clustering results generated from RNA-Seq, affymetrix by calculating the prediction probabilities of each the clustering results as the classification input with the corresponding data set using the *GaussianProcessClassifier()* function. After calculating the prediction probabilities, we relabeled the clustering probabilities using the *label.switching* package using the *stephens()* function. The function requires an $m \times n \times K$ dimensional matrix of allocation probabilities of the n observations, which are the unique cluster labels among the K mixture components corresponds to the both clustering results of both RNA-Seq and Affymetrix and for each iteration t = *1,…,m* which corresponds to the patient samples.

Recalling the main goal of the thesis, one of the main objectives is to aggregate the clustering results from multiple experiments. Once we obtained the label permutations, we changed the cluster probability labels from each cluster algorithms based on the results obtained by the *stephens()* function. For combining the cluster labels from each paired clustering algorithms, we implemented a naive Bayes approach multiplying the individual cluster allocation probabilities for each sample and then renormalise them so that the sum over k return to normal after renormalisation, accepting the cluster priors as identical.

$\forall k : P(k, A, B) = P(k \mid A) * P(k \mid B)$

$\forall k : P(k \mid A, B) = \frac{p(k,A,B)}{\sum_k P(k,A,B}$

After the cluster allocation probabilities aggregation, we obtained the cluster labels by returning the index of the maximum value over each row as the highest probability of the cluster labels. These cluster labels were then implemented for Differential Expressed Gene (DEG) analysis to apply the Gene Set Enrichment Analysis (GSEA).

Clustering validation results

Clustering validation has been acknowledge as an essential part to the achievement into clustering analysis. Recall the aim of clustering algorithms are to split data sets into classified objects such that the objects in the same group are similar as possible, and the objects in different clusters are highly distinct to each other.

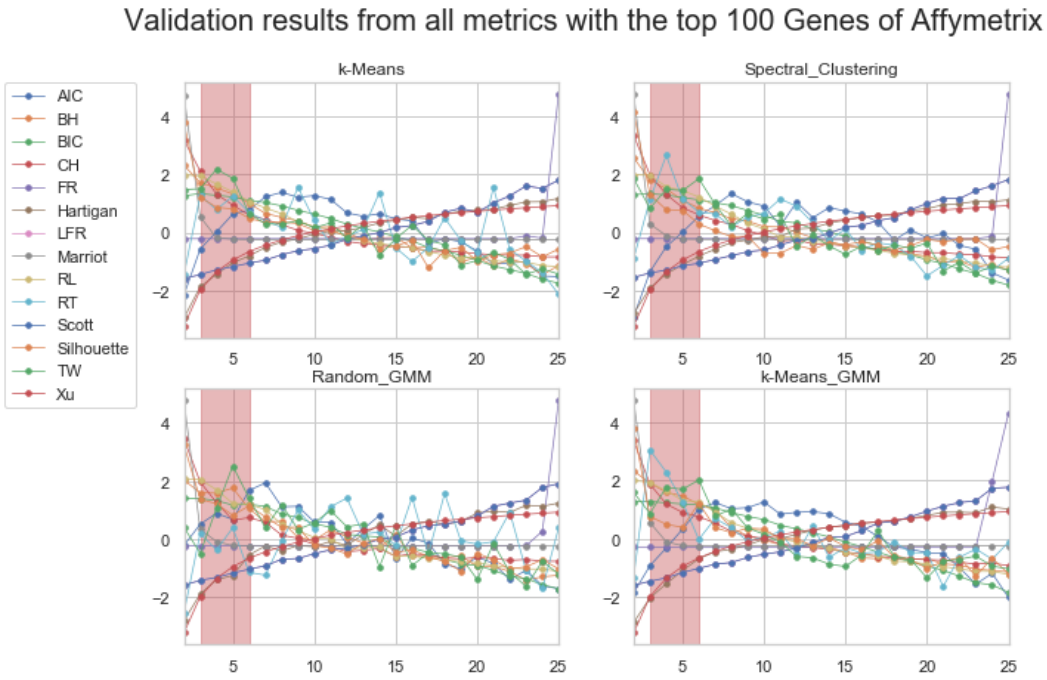

Figure 2.23: Clustering validation results using all available metrics with the top 100 ranked genes of Affymetrix data.

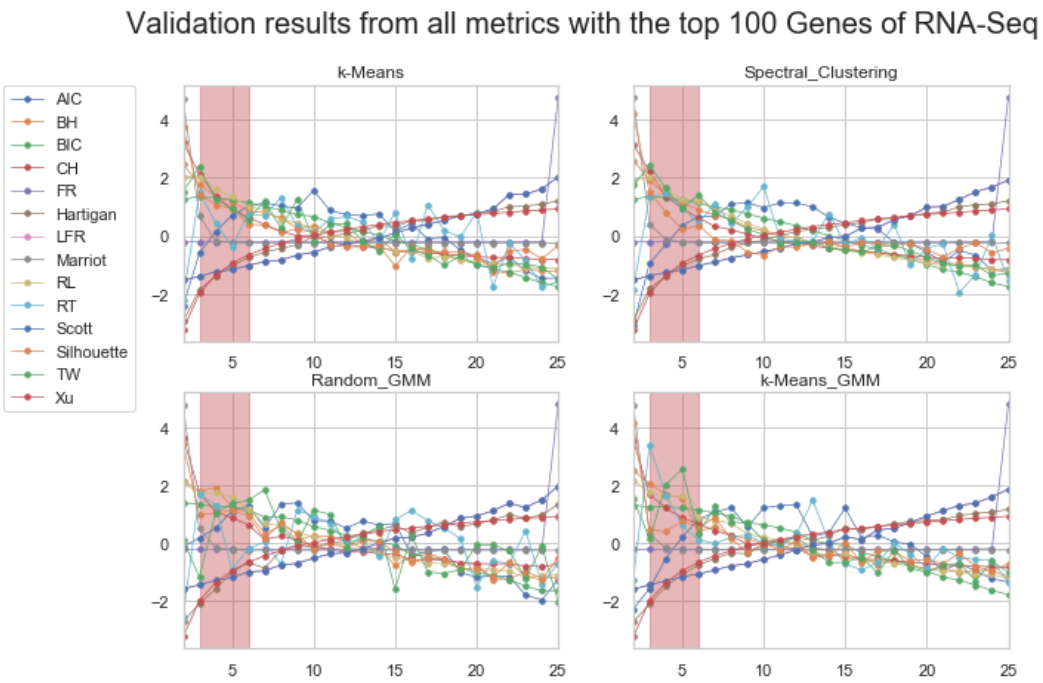

Figure 2.24: Clustering validation results using all available metrics with the top 100 ranked genes of RNA-Seq data.

The figures 2.23 and 2.24 illustrate the visualisation of the clustering validation results using all available clustering metrics. The red bar across cluster three to six shows the expectation range of the correct cluster number. At first no successful validation could be observed, and in attempt of visualising the clustering metrics in the same plot using a *StandarScaler()* function, no clear cluster number can be identified due to poor validation performance most of the clustering metrics produce. So we look into each metric separately to possibly detect any successful validation.

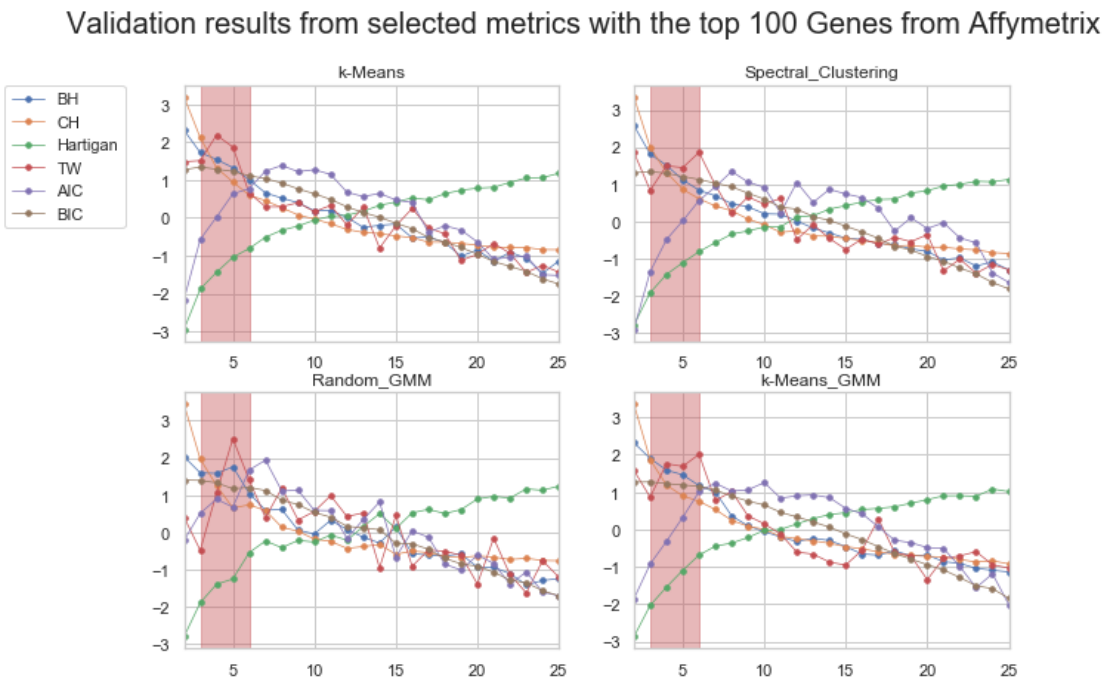

Figure 2.25: Clustering validation results of the selected metrics with the top 100 ranked genes of Affymetrix data.

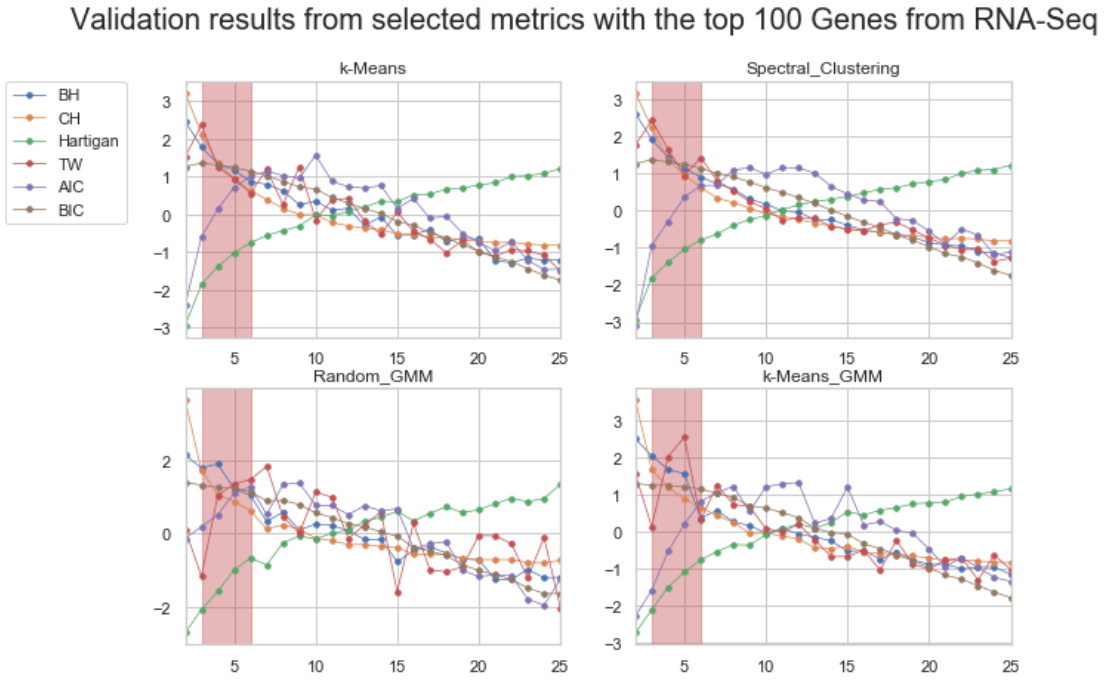

Figure 2.26: Clustering validation results of the selected metrics with the top 100 ranked genes of RNA-Seq data.

The figures 2.25 and 2.26 display the clustering validation of the selected metrics from the toy data set. Likewise to the figures 2.23 and 2.24 no clear clustering number could be classified. When retrospectively looking back on the validation results from the toy dataset, most of the metrics generate mostly persistent increasing or decreasing values after detecting the right number of clusters with barely no fluctuations. But we found a metric that show poor validation performance on the toy data set but looks promising for both Affymetrix and RNA-Seq data.

Marriott et. al. proposed an approach of identifying the correct cluster number by minimising the within-group dispersion matrix for data clustered into k groups(Marriott, 1971). The Marriot metric on the both figure 2.27 and 2.28 how a decreasing of the metric score as the cluster number increases. The elbow bend indicates that increasing the cluster number beyond cluster three have smaller values. The Marriot metric proposes that the optimal cluster number is three.

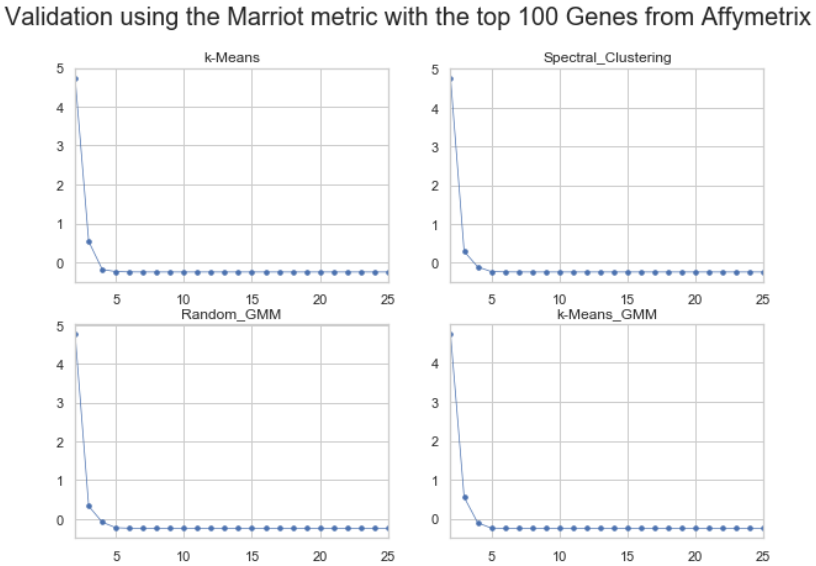

Figure 2.27: Clustering validation metric using Marriot of Affymetrix data with the top 100 ranked genes.

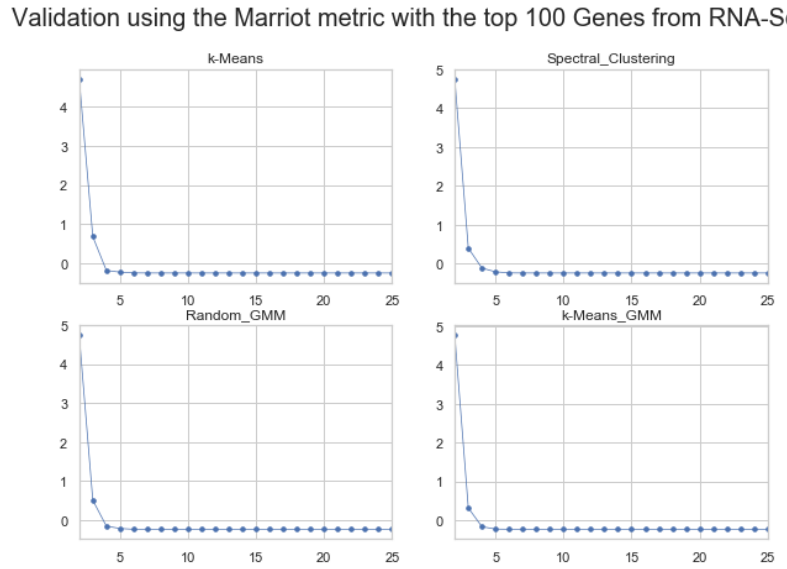

Figure 2.28: Clustering validation results of Affymetrix data with the top 100 ranked cancer genes.

We also implemented the Marriot metric to our mean aggregated data set. All cluster algorithms beside the random GMM show similar results to previous figures 2.27 and 2.28.

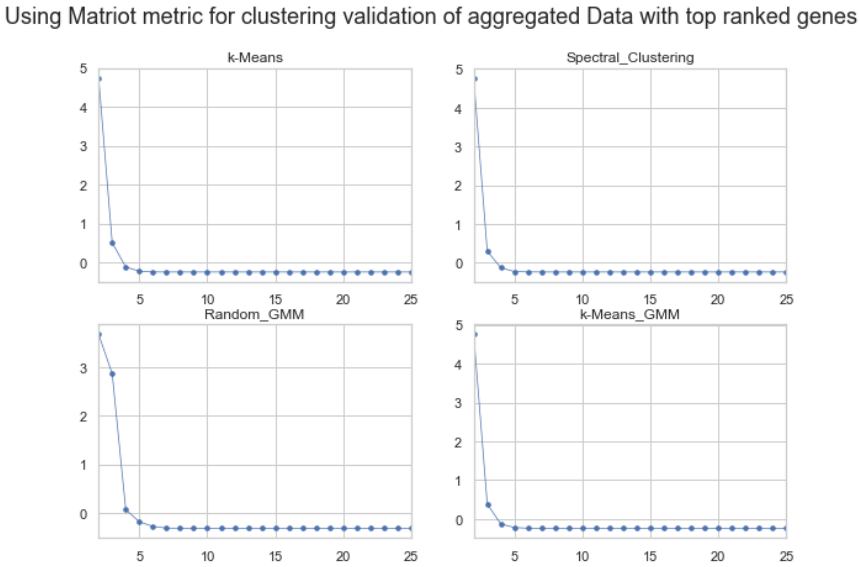

Figure 2.29: Clustering validation metric using Marriot of aggregated data with the top 100 ranked genes.

So overall, we used the *StandardScaler()* function from *sklearn* to unify different

clustering metrics and to look for patterns regarding visualisations in one graph for the optimal cluster number. Some clustering metrics are consistent throughout the clustering algorithms but some do predict different cluster numbers while few could not find any optimal cluster number at all. Although unifying different clustering metrics in the same graph can cause issues in obtaining a useful clustering validation, it is an important feature in a clustering validation approach. But one prominent validation metric displays a very appropriate clustering validation since it provides a very good example of an elbow method, which indicates cluster three as the optimal cluster number.

Clustering visualisation

In order to understand the extent of the results displayed from the previous chapters, we compared the clustering analysis results by projecting all data sets. It has to be noted that the figure presents all data sets in a 2-Dimensional form, transformed using t-SNE with default parameters from toy and perplexity of 450 and learning rate of 800 from both microarray data sets.

What makes t-SNE a good 2-Dimensional data visualisation tool is that it converts affinities of data points into probabilities. The affinities in the original space are represented by common Gaussian distributions and the affinities in the embedded space by t-distribution. Therefore, with the right parameter, t-SNE can separate groups based on their relative similarity by calculating the probabilities and the distance using KL-divergence between two data points. So what we should expect to see after applying t-SNE to the data, that the cluster labels should be at least in the same location close together, rather than spread around the graph.

Using a self designed toy data set will give us an understanding how the clustering algorithms work if the groups are clearly separated. The toy data set will then be compared as a reference to the clustering analysis results of our gene expression data.

Results from Toy data set

The figure 2.30 show a 2D generated toy dataset of 5 isotropic gaussians with 180 samples and 100 features and a standard deviation of 1.5 per gaussian with default

parameters. The dimension of the gaussians are designed in the way that it has the same dimension as the gene expression data. The toy dataset has been tuned to generate good clustering results. By setting the clustering results into a 2D projection, we can see visualise the performance of each clustering algorithm.

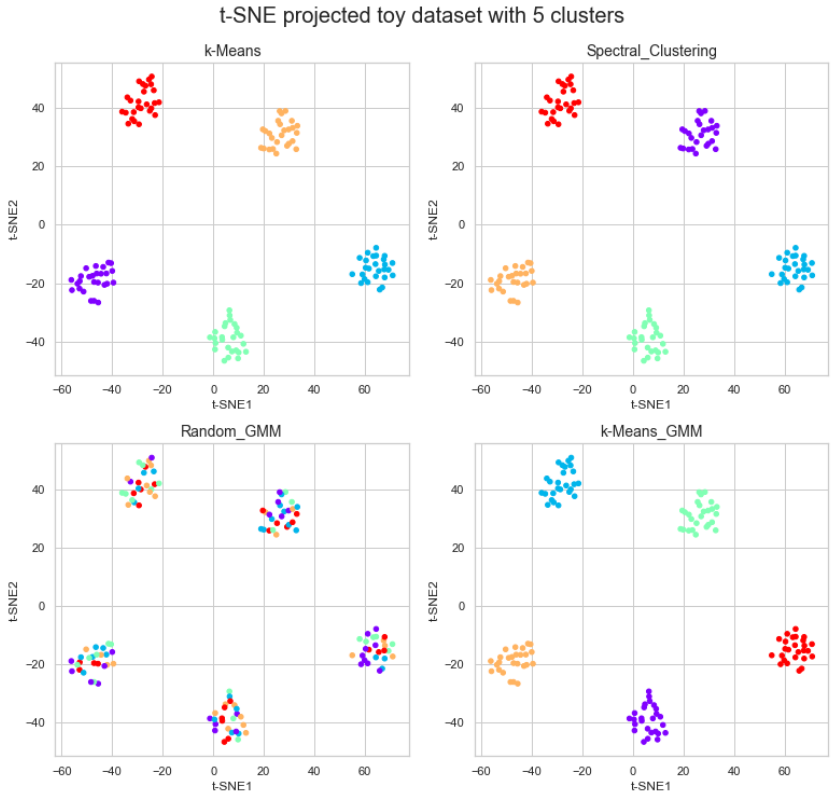

Figure 2.30: t-SNE projected toy dataset. The toy data set contains five isotropic gaussians. The different colour labels from each clustering algorithms, numerically labeled in each clustering analysis iteration is based on the so called label switching problem, which has been discussed on chapter 3.15.

The performance results of the clustering validation metrics can be clearly seen directly from the graphs. The toy data set has five well separated group values that can be easily be detected with the clustering algorithms, except for the random initialised GMM, all other clustering algorithms perform as expected with identical clustering results. The random GMM does not seem to clearly identify the five isotropic gaussians, with such a low standard deviation and the appropriate distance to each gaussians can be considered to be weighted less for the clustering analysis.

Results from the original, combined, and aggregated data

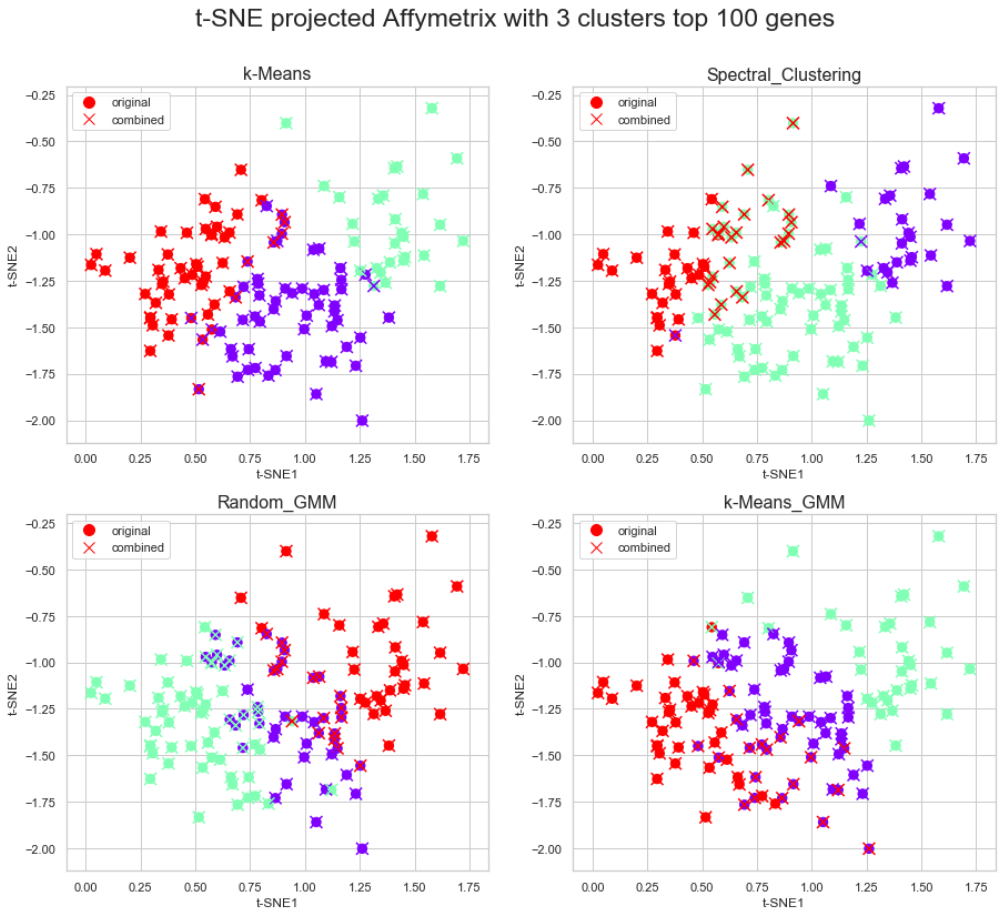

Figure 2.31: t-SNE projected of top 100 genes of Affymetrix data with three clusters. There are two different clustering results. The circles display the original clustering results while the x points are the results of the combined clustering labels from Affymetrix and RNA-Seq

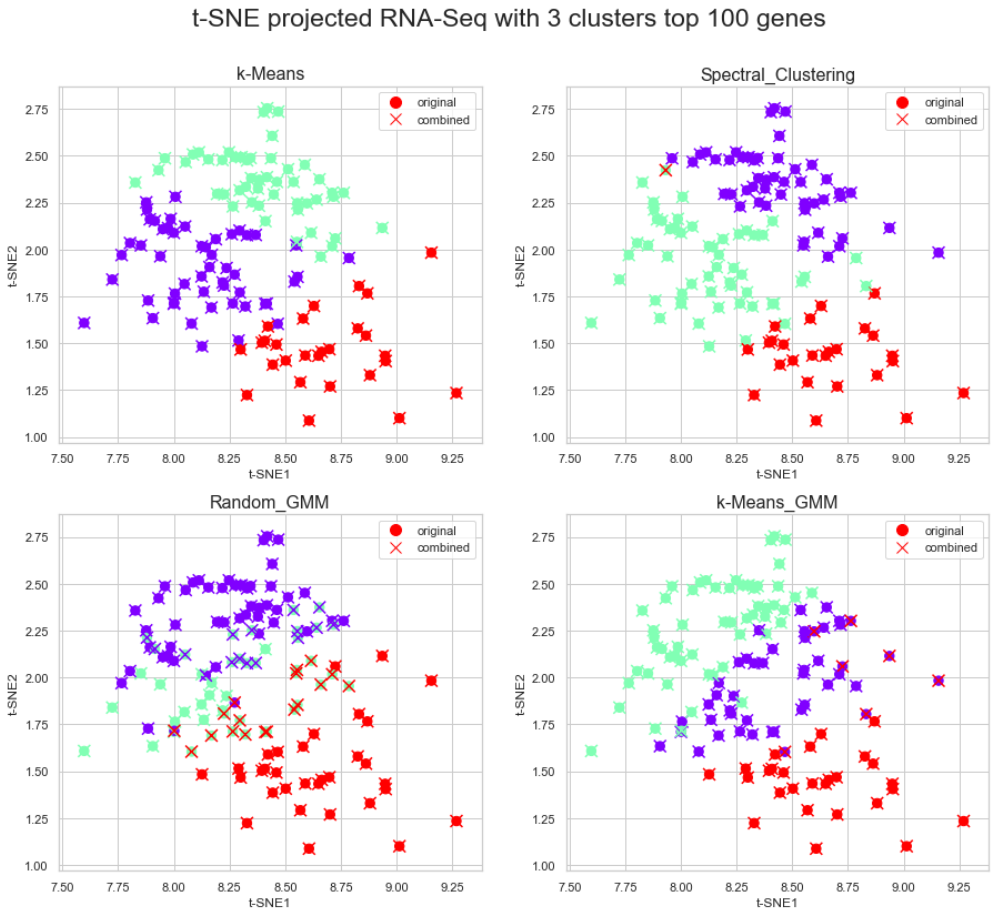

Figure 2.32: t-SNE projected of top 100 genes of RNA-Seq data with three clusters. There are two different clustering results. The circles display the original clustering results while the x points are the results of the combined clustering labels from Affymetrix and RNA-Seq.

In this section we show a scatter plot from Affymetrix and RNA-Seq. The x represents the original labels of the clustering results, and the circle points represents the labels from the combination of both clustering results by the Naive Bayes approach. Both data were projected with $n = 2$ components and *perplexity* $= 120$. Each color defined as the cluster label resulted from the clustering analysis. Both data display good clustering classification results of the three clustering groups on all algorithms. The clustering analysis results generated by RNA-Seq show high agreements on k-Means, Spectral Clustering, and k-Means GMM, although random GMM just vary slightly from the others. Whereas Affymetrix, all four cluster algorithms produced mostly identical clustering results. Similar clustering labels from all algorithms indicate that cluster three as the correct number of clusters.

In the case of RNA-Seq data, no major changes in the cluster labels could be observed after the clustering combination on k-Means und Spectral Clustering, while few patients were switched over from one group to the other on k-Means GMM. On one group of random GMM, many patients have switched over the other two groups. On Affymetrix data, only few of the patients from two groups (purple and red) are switched to each other. Many patients have switched from one group to the other group (turquoise to red on Spectral Clustering, purple to turquoise on Random GMM, and purple to red on k-Means GMM).

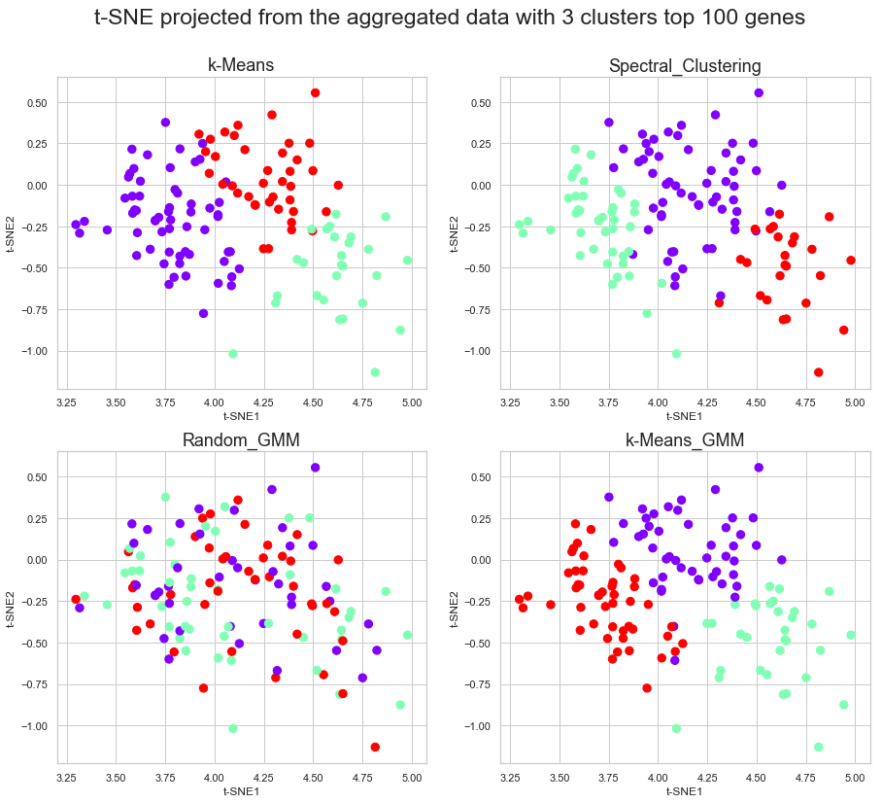

Figure 2.33: t-SNE projected of top 100 genes of the mean aggregated data with three clusters.

The figure 2.33 shows the t-SNE projected clustering visualisation of the aggregated data. Similar to the results generate from both gene expression data, the aggregated data appear to also have agreements on clustering results when comparing visually on all clustering algorithms beside from the random GMM, similar to the clustering results visualisation from generated from the toy data set. All three clustering algorithms (k-Means and k-Means GMM) show a slightly more similar clustering labels

than Spectral Clustering.

t-SNE is a method well-suited for visualising of multidimensional data in a low 2-dimensional space, using the local relationships between data points. t-SNE creates a probability distribution using Gaussian distribution that defines the local relationships. Compare to PCA which maps high-dimensional spaces around the medium distance, which makes the distances between the low-dimensional points gather around the medium distance that can cause to the 'crowding problem'. t-SNE deals with the problem by spreading out the medium distance points with certain parameters to prevent the crowding. Hence, it takes similar data points and place it on similar spaces separated from the other data points. Therefor, t-SNE confirms that the algorithms have generated good clustering results with the optimal cluster number of three. Based on the results from the toy data and the mean aggregated data sets, we should take the results generated by the random initialised GMM less into account.

Clustering similarity results

Figure 2.34: Clustering similarity results of Affymetrix (left) and RNA-Seq (right) of the 100 top ranked genes.

The figures 2.34 displays the clustering similarity results using Fowlkes-Mallows score that compares the clustering algorithms with each other. The results indicate that the higher the cluster number the lesser the agreement of the clustering analysis results. Based on the higher correlations presented on the figure 2.21 between Affymetrix and HTSeq, we actually expected an identical similarity results between both gene expression.

On the Affymetrix (left), all clustering algorithms show very high similarity > 0.90 comparing with each other with Random GMM vs. k-Means GMM as the highest with identical clustering results (score = 1). On cluster five, the similarity for all algorithm is lesser compare to cluster six, that indicate an agreement that cluster five is not the optimal cluster number when choosing between one to six. After cluster six, the scores show a decreasing trend with some exceptions at certain cluster numbers. But generally the similar decreases as the cluster number increases, with difficulties for the algorithms to find more alignments.

RNA-Seq significantly show lesser agreements on similarities compared to Affymetrix, but a visible indication of the highest similarity scores on all algorithms around cluster three and the decreasing similarities beyond cluster three. The highest similarity score is about 0.89 (k-Means vs. Spectral Clustering) on cluster two followed on cluster three of about 0.86. When looking at cluster two, we can see that Random GMM show the least agreement against all other clustering algorithms, which means a poor clustering analysis performance.

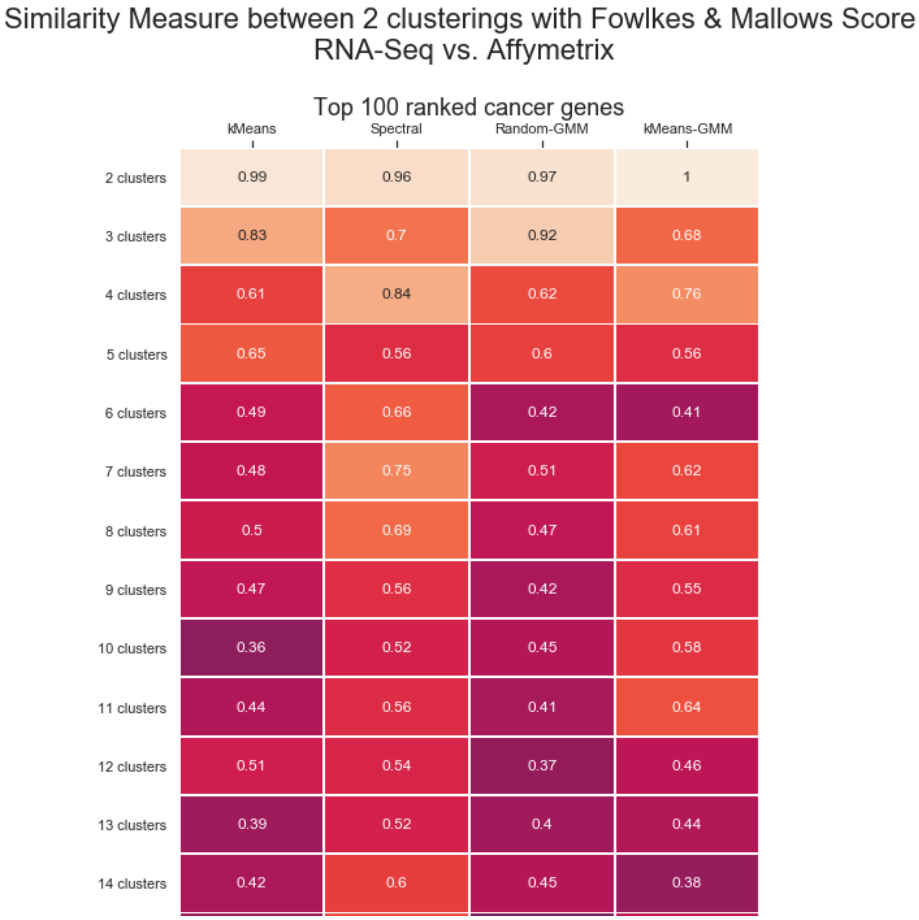

Figure 2.35: Clustering similarity results of Affymetrix vs. RNA-Seq from both the 100 top ranked genes.

On the figure 2.35 the similarity score with matched algorithms on both gene expression data with, were being calculated. Inconsistencies of clustering agreements can be demonstrated between cluster two and five, with an exception of Spectral Clustering, which show a general higher similarities along the cluster numbers compare to the other algorithms. We conclude that the results using the similarity score of Fowlkes-Mallows reinforces the results generated from our cluster validation, as we can see that the scores beyond cluster three degrade the higher the cluster number get.

Survival analysis: Investigating clustering specific phenotypes

To assess whether the clustering leads to clinically relevant patient groups multivariate survival analysis via Kaplan-Meier has been conducted. We took the phenotypic data with all the meta data including the time periods to death. We then annotated the phenodata with its corresponding clustering labels (the combined and the aggregated cluster labels) and perform a Kaplan-Meier survival analysis using the *lifelines*(Davidson-Pilon et al., 2019) library. The *KaplanMeierFitter()* is the object to conduct the survival analysis if each cluster. We further conducted a multivariate Log rank test using the *multivariate_logrank_test()* to calculate the difference between the cluster groups per algorithm and the similarities per group across the algorithms under the null hypothesis of the groups are similar to each other and the alternative hypothesis of at least one groups that differs from the other groups.

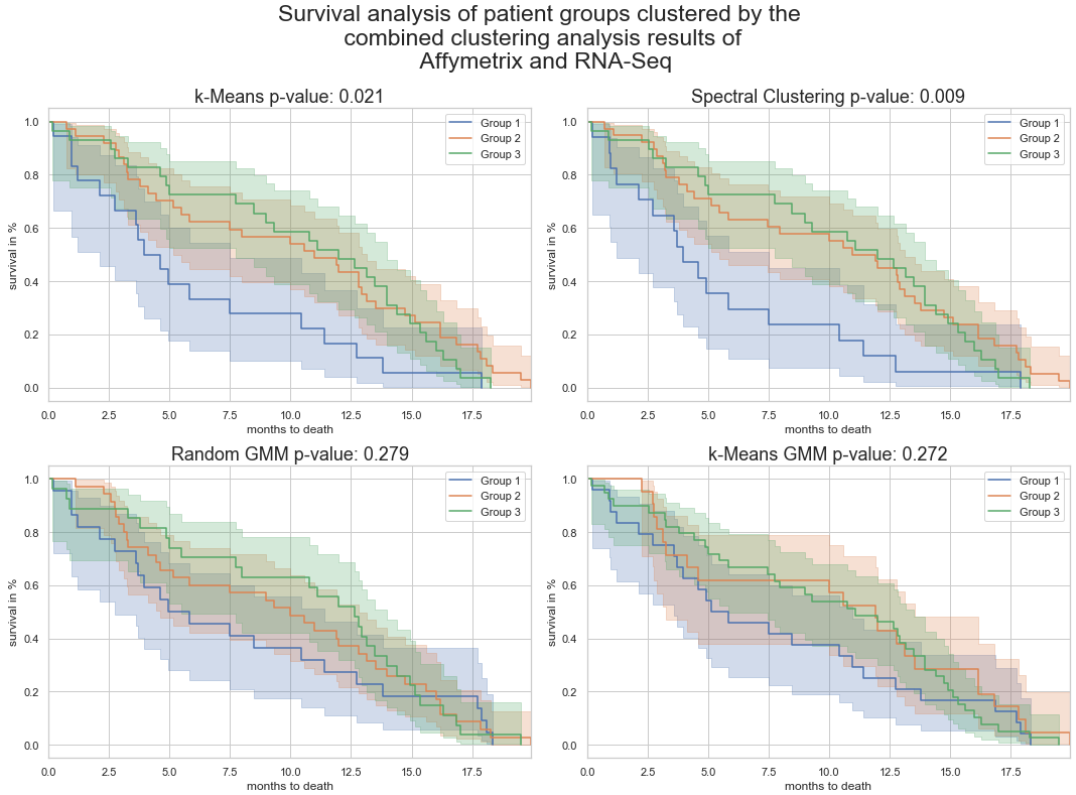

Figure 2.36: Survival analysis with Kaplan-Meier Method of the combined cluster labels of each clustering algorithm.

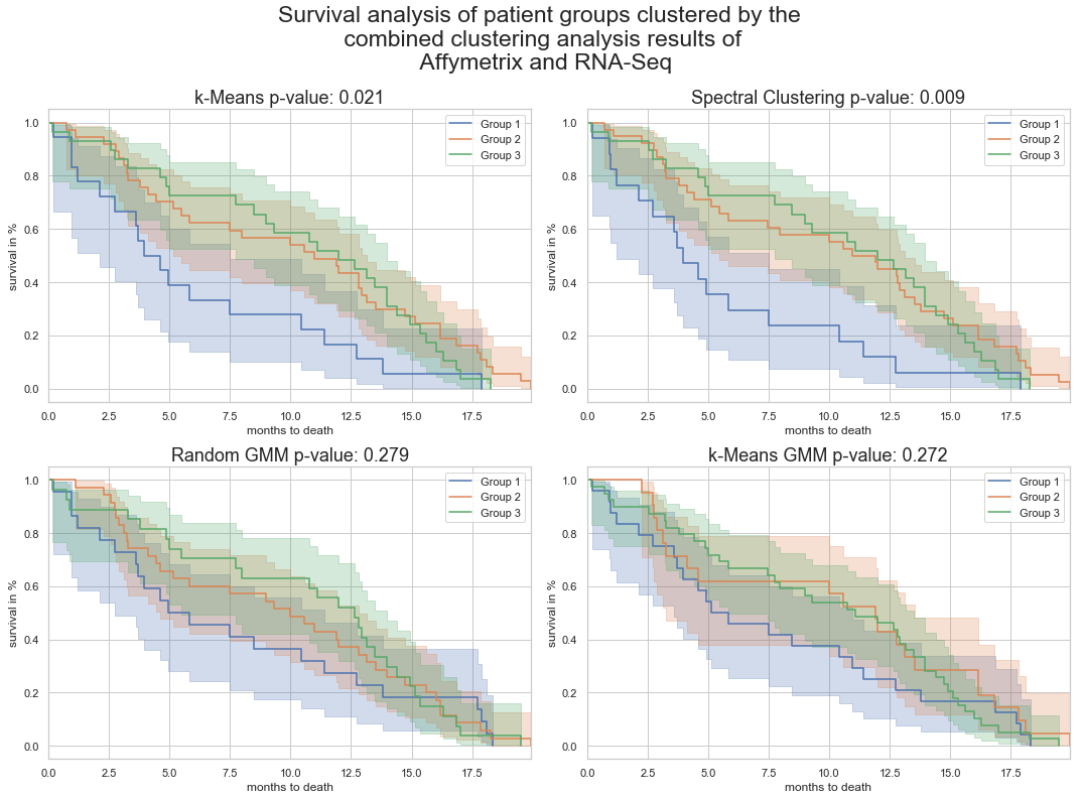

Figure 2.37: Survival analysis with Kaplan-Meier Method of the aggregated cluster labels of each clustering algorithm.

The figures 2.36 and 2.37 show the depiction of a survival analysis according to Kaplan-Meier. Each plot represents each cluster algorithm with 3 different cluster groups, representing the survival probability of each individual group as a cumulative distribution function. The log rank p-value is also shown in each plot. k-Means (p-value = 0.021) and Spectral Clustering (p-value = 0.008) show significant different between all three groups with Spectral Clustering as the most significant with big differences between the groups, while either one or more clusters of Both Gaussian Mixture Models (random GMM = 0.279, k-Means GMM = 0.272) do not significantly show any differences between the groups on the combined cluster results 2.36. In the case of the aggregated cluster labels 2.37, only Spectral Clustering (p-value = 0.008) displays significant differences between groups in terms of survival probabilities.

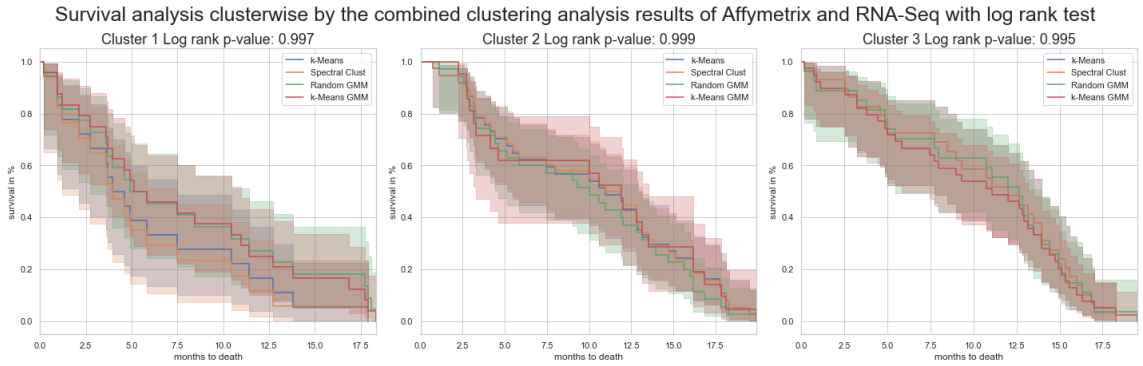

Figure 2.38: Survival analysis with Kaplan-Meier Method of the combined clustering results between the clustering algorithms with each group.

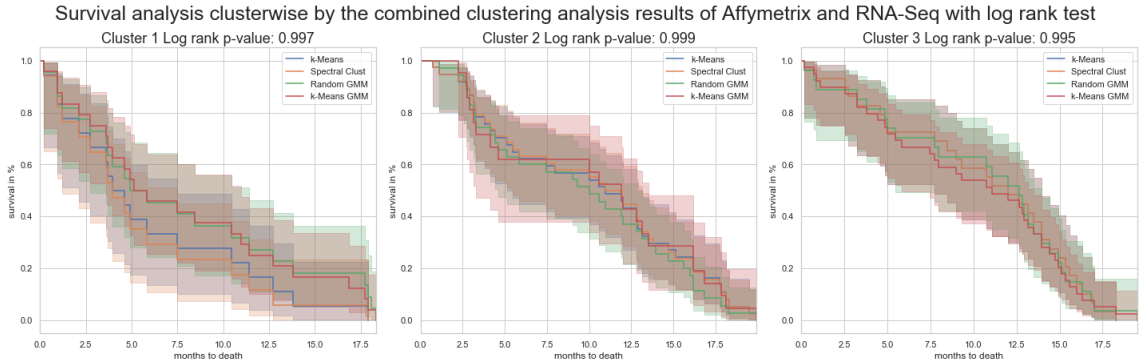

Figure 2.39: Survival analysis with Kaplan-Meier Method of the aggregated cluster labels of each clustering algorithm.

We additionally applied the multivariate log rank test to determine the similarities of the groups between the cluster algorithms or differences between cluster groups and visualise the survival curves represent by the figures 2.38 and 2.39. Regarding the survival analysis results of the combined cluster labels, all groups have obtain very high p-values close to 1, indicating high similarities at least two of the clustering algorithms in corresponding to the other algorithms (cluster 1: p-value = 0.997, cluster 2: p-value = 0.999, cluster 3: p-value = 0.995). Comparable are the results of the aggregated cluster labels, where all the algorithms in each group have high concordance, with cluster 1 has the highest similarities and cluster 2 as the lowest (cluster 1: p-value = 1.0, cluster 2: p-value = 0.843, cluster 3: p-value = 0.968).

This section describes the results using Kaplan-Meier plots to visualise survival curves to each groups and Log-rank test to compare the survival curves of these groups. The

results suggests that grouping patients by expression data clustering separates patients with short to medium survival in three distinct groups which show (significant) differences in survival.

DEG and GSEA results

Differential Expression Analysis

We conducted a Differential Expressed Genes (DEG) analysis based on the generated clustering results to each corresponding clustering algorithms in each gene expression data sets, with the purpose of discovering quantitative changes in expression levels between cluster groups. We used the *limma* package for calculating DEG, which generated p-values per genes. Each DEG analysis results were further implied separately for the GSEA. As a results, gene list has been generated with p-values annotated to its corresponding gene symbol.

The figure 2.40 shows logit transformed p-values compared with each cluster algorithm between Affymetrix and RNA-Seq. As you can see in the graphs, there is clearly positive correlation between the two datasets, in spite of the scatter being more spread out. Also the scatter plot between each of the clustering algorithms are very similar.

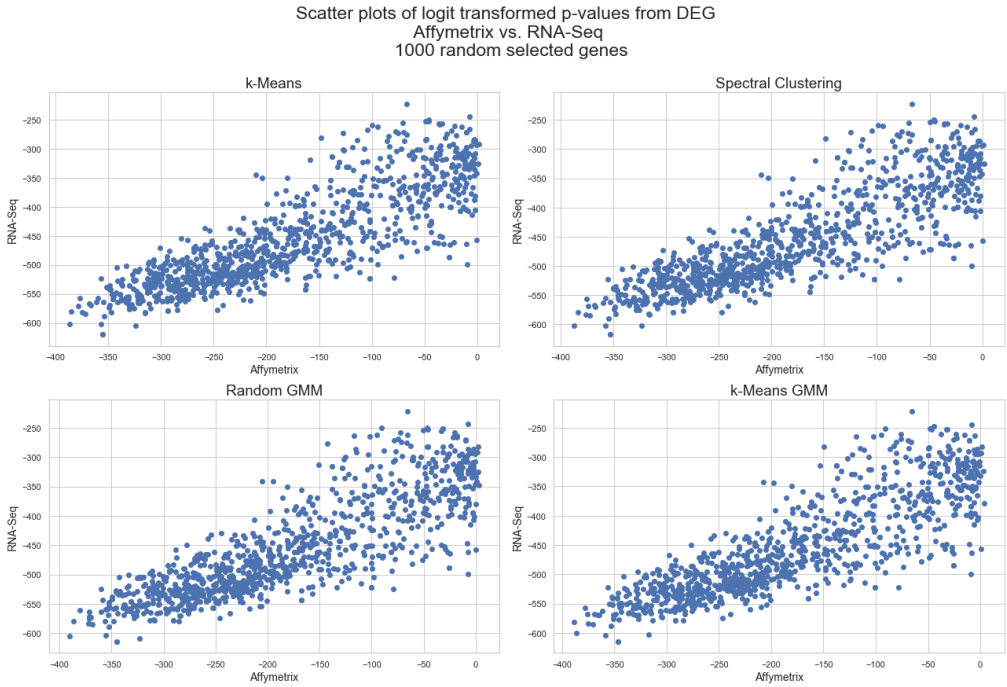

Figure 2.40: Scatter plot of logit transformed p-values Affymetrix vs. RNA-Seq compared to each paired clustering algorithms of 1000 random selected genes.

For DEG analysis, we transformed our gene expression data frame into an *ExpressionSet* by importing the *phenoData* annotating each sample with additional meta data, and *featureData*, which annotate the gene symbols with additional information such as ENSEMBL IDs, and gene description. DEG analysis with *limma* also requires beside the expression set a design matrix, composed of the combined cluster labels converted as factors. Subsequently, fitting a linear model with the expression set and the design matrix as inputs, calculating the empirical Bayes Statistics for differential expression of the standard errors towards a common value, given a linear model fit. As results, we get the moderated t-statistics, F-statistics, and log-odds of differential expression. We then take the p-values corresponding to F-statistics as our p-values for our GSEA.

Gene Set Enrichment Analysis

In this section, we discuss the application of Gene Set Enrichment Analysis (GSEA) and the corresponding results to identify classes of genes that are over-expressed in

our data, that may reveal associations with GBM cancer.

Once we have the gene list with their corresponding *p*-values generated from the DEG analysis, we also require the gene annotations and the *annFun.db* function is used to extract the gene to Gene Ontology annotation from the *affyLib* objects. Thus, we also have to define a function as a gene selection criteria to out enrichment analysis, in our case selected the top 10% of most significant genes instead of a regular *p*-value cutoff. We have now all the necessary data and functions to build the object required for the *topGOdata*.

We then continued to conduct the enrichment analysis, we used the Kolmogorov-Smirnov (KS) test which computes the enrichment based on gene scores, mentioning the *nodeSize = 10* for the GO terms hierarchy to show less than 10 gene annotations. The function *runTest()* is implemented to apply the KS statistic and the method of the *topGOdata*. The function returns a *topGOresult* class. *GenTable()* is a function for returning the most significant GO terms and the corresponding *p*-values into a readable data frame. *showSigOfNodes()* is a function which visualises the enrichment analysis over a Gene Ontology graph.

The figures below display the Gene Set Enrichment Analysis Results of Affymetrix, RNA-Seq and the mean aggregated data. Since all four clustering algorithms to each data sets generate almost identical GSEA results, we only display the GO graph results from the DEG analysis of all data set with k-Means cluster labels of Affymetrix.

The figures display Gene Ontology graphs induced by the top 5 most significant Gene Ontology terms. Significant nodes are represented as rectangles, ranking from the most significant (dark red) to the least significant (bright yellow). For each node, the GO identification number and the *p*-values are displayed. Additionally to the GO graph, we displayed the summary table with the results from the test using the Kolmogorov Smirnov method. The table contains the top nodes of GO terms ordered by the *p*-values.

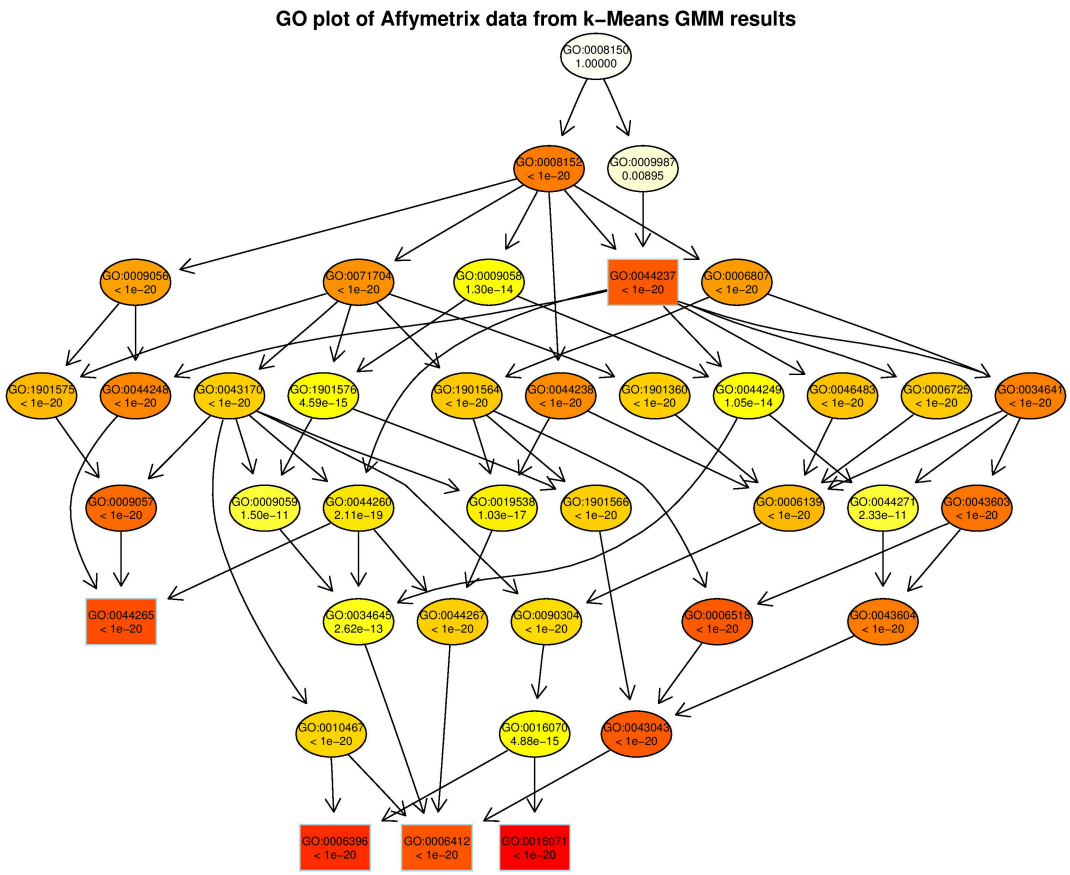

Figure 2.41: Gene Ontology graph of the top 5 GO terms identified by the KS (Kolmogorov-Smirnov) algorithm based on DEG analysis results of Affymetrix with k-Means cluster labels.

	GO.ID	Term	Annotated	Significant	Expected	KS
1	GO:0016071	mRNA metabolic process	543	141	54.31	< 1e-30
2	GO:0006396	RNA processing	608	117	60.81	< 1e-30
3	GO:0044265	cellular macromolecule catabolic process	764	187	76.41	< 1e-30
4	GO:0006412	translation	418	128	41.80	< 1e-30
5	GO:0044237	cellular metabolic process	6263	712	626.37	< 1e-30
6	GO:0006518	peptide metabolic process	531	149	53.11	< 1e-30
7	GO:0006886	intracellular protein transport	794	187	79.41	< 1e-30
8	GO:0043043	peptide biosynthetic process	434	130	43.40	< 1e-30
9	GO:0009057	macromolecule catabolic process	919	202	91.91	< 1e-30
10	GO:0022613	ribonucleoprotein complex biogenesis	321	73	32.10	< 1e-30
11	GO:0006413	translational initiation	145	72	14.50	< 1e-30
12	GO:0043603	cellular amide metabolic process	694	162	69.41	< 1e-30
13	GO:0043604	amide biosynthetic process	521	138	52.11	< 1e-30
14	GO:0008152	metabolic process	6609	732	660.97	< 1e-30
15	GO:0000375	RNA splicing, via transesterification re...	252	68	25.20	< 1e-30
16	GO:0006397	mRNA processing	345	78	34.50	< 1e-30
17	GO:0000377	RNA splicing, via transesterification re...	249	68	24.90	< 1e-30
18	GO:0000398	mRNA splicing, via spliceosome	249	68	24.90	< 1e-30
19	GO:0046907	intracellular transport	1276	247	127.61	< 1e-30
20	GO:0034641	cellular nitrogen compound metabolic pro...	3965	470	396.54	< 1e-30
21	GO:0008380	RNA splicing	305	74	30.50	< 1e-30
22	GO:0044248	cellular catabolic process	1469	255	146.92	< 1e-30
23	GO:0044238	primary metabolic process	6149	685	614.97	< 1e-30
24	GO:0071704	organic substance metabolic process	6367	702	636.77	< 1e-30
25	GO:0006613	cotranslational protein targeting to mem...	76	50	7.60	< 1e-30
26	GO:0006605	protein targeting	297	100	29.70	< 1e-30
27	GO:0006807	nitrogen compound metabolic process	5908	661	590.86	< 1e-30
28	GO:0045047	protein targeting to ER	81	52	8.10	< 1e-30
29	GO:0072594	establishment of protein localization to...	401	114	40.10	< 1e-30
30	GO:0070727	cellular macromolecule localization	1296	253	129.61	< 1e-30

Figure 2.42: Summary table with the results corresponding to the GO graph from k-Means clustering labels of Affymetrix.

The figure 2.42 displays a summary of the most significant GO terms ranked by the corresponding p-values from KS.

We are going to discuss the most significant GO terms (the rectangles) into our discussion, to find out if these GO terms have any relevance in Glioblastoma Multiforme. The top GO terms which are matched to all data sets are GO:0006396, GO:0044237, 0016071, and 0008152 which are annotated to RNA process, cellular metabolic process, mRNA metabolic process and metabolic process.

GBM is the most common type of primary brain cancer and for most GBM solid tumours there are currently no effective treatments available besides the current surgical resection, followed by radiotherapy with temozolomide (TMZ). Although, current advanced therapies such as gene therapy, and immunotherapy are in clinical trials, the survival rate of GBM patients barely improved over the last decades. Therefore, new treatment methods are urgently on demand. Researchers published a paper which proposed that RNA-Processing as a therapeutic route for GBM treatment.

Alternative splicing is an importance source for gene regulation that affects more than 90% of the human coding genes and that mutations and alterations in splicing factors that display potential tumour driving mechanisms(Meliso et al., 2017). Thus, mutations and alterations of splicing factors are prawn to induction of genomic instability, which is a common characteristic in GBM. RNA-processing regulation occurs by a complex RNA-protein network and changes in the regulation can cause to mutations and splicing aberrations and hence to cancer(Yeo, 2016).

They stated that since defects in mRNA splicing can cause a high probability of developing cancer, splicing modulation has been a promising approach in GBM treatment(Meliso et al., 2017). Metformin, together with TMZ has the task to inhibit the proliferation of GBM cells. They also mentioned that based on their gene expression profiling results, metformin has been involved in RNA binding and splicing(Meliso et al., 2017).

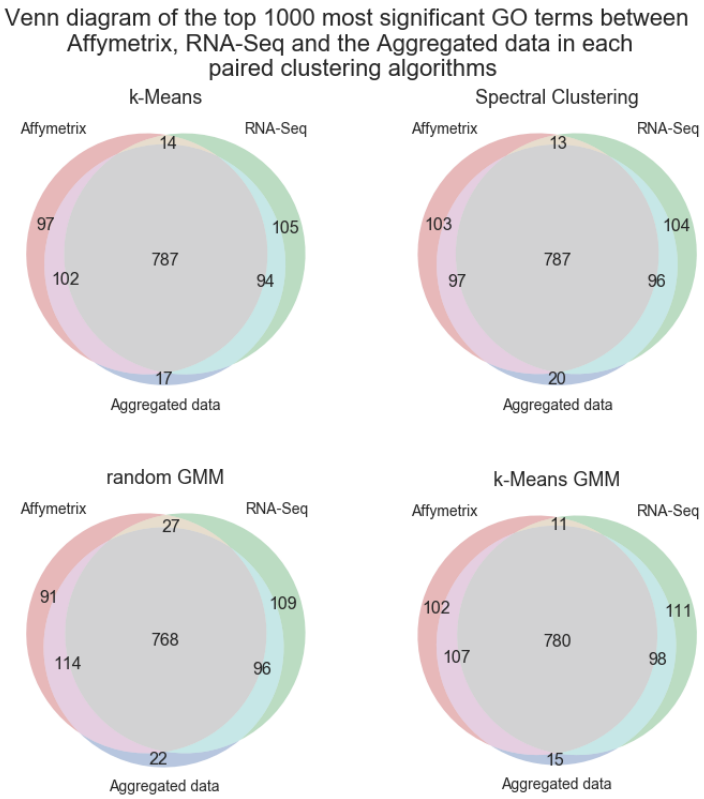

Figure 2.43: Venn diagram of the Gene Set Enrichment Analysis results from Affymetrix, RNA-Seq and the aggregated data set.

The figure 2.43 displays a Venn diagram of the top 1000 GO terms resulted from the enrichment analysis of all gene expression data. The Venn diagram serves as an illustration of the enriched GO terms provided by the GSEA analysis of all three data. There are 7 regions in the Venn diagram in each cluster algorithm which show the intersections and difference between GO terms of the data. k-Means clustering shows that 787 significant GO terms are matched between all three gene expression data. 97 GO terms of Affymetrix, 105 of RNA-Seq and 17 of the aggregated data neither intersect, hence, not enriched from each other. 13 GO terms which are enriched in both Affymetrix and RNA-Seq are not enriched in the aggregated data. All other algorithms display similar results to the k-Means Venn diagram with very small variations of GO terms across the intersections.

To validate our GO enrichment analysis results, we additionally conducted a GO terms comparison to other similar published articles. We collected the top GO terms of those randomly selected studies to verify the existence of these GO terms to the

top 1000 GO terms of our enrichment results. The following selected studies are:

- Predicting glioblastoma prognosis networks using weighted gene co-expression network analysis on TCGA data(Xiang et al., 2012)
- Identification of glioblastoma gene prognosis modules based on weighted gene co-expression network analysis(Xu et al., 2018)
- Identification of hub genes and pathways in glioblastoma by bioinformatics analysis(Yang et al., 2019)
- Differential gene expression analysis in glioblastoma cells and normal human brain cells based on GEO database(Wang and Zhang, 2017)
- Discovery and validation of a glioblastoma co-expressed gene module(Dunwoodie et al., 2018)

The study published by Xiang et. al.(Xiang et al., 2012) and Xu et. al.(Xu et al., 2018) utilised similar methods with the goal constructing a gene co-expression network in TCGA GBM samples and to conduct enrichment analysis to classify significant gene modules. Based on our top 1000 GO terms, we could find more agreements to Xiang's GO enrichment analysis results (6 out of top 14 GO terms) compare to Xu's results (1 out of top 10 GO terms) displayed in the studies. The most matched GO terms were found so far on the article published by Yang et. al.(Yang et al., 2019), with the scope of understanding the molecular mechanism of GBM to provide novel treatment strategies (10 out of 21 top GO terms listed in the publication). The study published by Wang et. al.(Wang and Zhang, 2017), conducting a DEG analysis between GBM cells and healthy human brain cells based on GEO database. Likewise we could find GO terms from their results, that also correspond to the top 1000 GO terms we generated from the GO enrichment analysis (6 out of top 11 GO terms). And the last study we compared was conducted by Dunwoodie et. al. with the scope similar to the studies of Xiang et. al.(Xiang et al., 2012) and Xu et. al.(Xu et al., 2018), utilising a gene co-expression network analysis. We only found 1 matched GO term from 11 published GO terms.

We further looked into the cell biology and the intracellular events that characterises GBM and then compared these events with our GO terms. Based on the review of (Nakada et al., 2011) these events arise mostly In combination of multiple tumorigenic events.

The first biological event he stated is about the loss of cell cycle control. GBM develops different ways to evade the cell cycle control for its own growth benefits. G1-S phase transition is one of the cell cycle point that obtained the most attention in the cell cycle event of GBM.The second event is the over-expression of growth factors and receptors, such as epidermal growth factor receptor (EGFR), platelet-derived growth factor (PDGF),Transforming growth factor (TFG) and fibroblast growth factor are over-expressed In GBM. He also mentioned the *p53* pathway of GBM that cause abnormality of apoptosis by disturbing the apoptotic response that follows the usually strict control of cell cycle progression, as well as genetic instability, which encourages further genomic damage and mutations in *p53* would lead to genomic instability and hence to tumour progression.

All these biological processes are important indications of GBM. A manual investigation of GO terms which we find by the proposed GSEA as significantly enriched reveals GO biological process terms like GO:0045786 - negative regulation of cell cycle, GO:0044843 - cell cycle G1/S phase transition, GO:1902806 - regulation of cell cycle G1/S phase transition, GO:0008543 - fibroblast growth factor receptor signaling pathway, GO:0072331 - signal transduction by p53 class mediator. By linking the gene expression signature which differentiates the three proposed clusters with known biological implications of GBM development we obtain additional confirmation for the proposed approach to uncover GBM subtypes by an unsupervised analysis of expression data. # Conclusion and Future Work

We explore in this MSc thesis whether a robust method for identifying subgroups of GBM patients can be obtained by aggregating multiple data sets into a unified clustering. The clustering uses carefully normalised and selected gene signatures to identify a grouping of GBM cases which solely relies on expression signatures. Cases which are assigned to the same cluster share similar expression patterns while differing from other patient groups. Such clustering leads to robust expression signatures for characterising GBM subtypes. The acquired knowledge has thus great potential to provide reproducible GBM subtypes which may subsequently improve diagnosis and treatment of GBM. The gene expression datasets we used in this MSc thesis are Affymetrix microarrays and HTSeq-counts of mRNAs on gene level. Agilent gene expression data that would have been available as well have were excluded for reasons of low correlation with these two data modalities. To avoid methods dependencies

and to assess robustness across clustering techniques we used 4 different algorithms including k-Means, Spectral Clustering and Gaussian Mixture Models that were initialised randomly and by k-Means.

For data aggregation we applied two different methods:

1. Expression data based aggregation combined HTSeq and Affymetrix data after pairing samples and transcript clusters and separate data normalisation. To adjust for platform specific effects in gene expression quantification we adjusted the data by the standard scaler function before taking average expression as summary quantification of both platforms. Since we necessarily have to reduce samples and genes to those found in both platforms, analysis on aggregated measurements have to discard information which if avoided could have lead to an improved clustering.

2. Combining clusterings at the meta level of cluster assignments allows us to use platform specific variables for clustering. As cluster analysis is done for all platforms separately, we may all samples and thus more information to arrive at a more reliable clustering of the data. If we wish to use the same input variables, we may use Fishers meta-analysis to obtain a shared ranking without need to pool the data. Agreement in samples is only required for identifying cluster centres across different input modalities and clusterings. The GBM subtypes obtained with the second proposition is thus based on more information and should thus be more robust.

Clustering plays an important role in both data integration strategies. A critical aspect of clustering is to determine an appropriate number of clusters. This is technically challenging and from the biological perspective the most crucial aspect as it determines the number of GBM subtypes. The number of clusters is biologically important as it determines the number of proposed GBM subtypes. Determining the optimal number of clusters is moreover technically difficult as model fitting will always prefer larger cluster numbers and additional penalties must be considered. To get reliable estimates for the optimal number of clusters, we apply several metrics that were on toy data found to provide reasonable predictions. The proposed number of three GBM subtypes results from consensus considerations across all metrics. Attempting to cluster the data into a larger number of subtypes led to a degraded

reproducibility and was thus considered inappropriate.

A careful validation suggests that both aggregation methods generate robust clusterings. We find however that random initialised Gaussian Mixture Models are more difficult to fit and do not always provide meaningful results. This could however also be a shortcoming of the GMM implementation in the toolbox that was used for our assessments. Judged by cluster robustness we find that clustering of multivariate GBM expression profiles with medium sized gene numbers leads after cluster identification with (Stephens, 2000) to large agreement in cluster assignments independent of the chosen method. Cluster methods are nonlinear and have many local optima in parameter space. Although optimisation is challenging, we find that repeating the optimisation often enough from different starting conditions leads to reproducible results. We may thus conclude that a careful machine learning workflow leads to a separation of TCGA-GBM cases on the basis of expression signatures into three distinct subtypes.

An important prerequisite for putting forward characterisations of diseases like GBM is to assert that proposed subtypes which follow from purely technical consideration are also biologically meaningful. To provide such verifications, this thesis uses two different assessments.

1. We used a pairing of expression data with clinical information to tag all cases with patient survival. This allows us by a Kaplan-Meier analysis (Kaplan and Meier, 1958) to assess our clustering based GBM subtypes for significant differences in patient survival. If we focus on a medium survival interval which contains a large fraction of cases, we find that patient survival depends significantly on the cluster assignment. Despite that not all clustering methods find significant associations this analysis suggests that the proposed clustering uncovers expression patterns which are linked to GBM aggressiveness and proves that our propositions are viable.
2. To provide a second assessment of biological relevance, we used the assigned clusters as rank levels in a limma analysis. Easy to comprehend biological meaning is subsequently obtained by subjecting the resulting gene list to a GSEA analysis (Alexa and Rahnenführer, 2009) where we used the Gene Ontology as tag terms. A comparison with published GBM analyses which use the same evaluation strategy reveals great overlap among top ranked GO terms of up to

50%. A careful manual comparison of our GO term list with a biological characterisation of GBM that was reported in (Nakada et al., 2011) also uncovers considerable overlap in GBM linked molecular mechanisms. Among the biological processes and pathways which are defined by our clustering and linked with GBM we find: GO:0045786 - negative regulation of cell cycle, GO:0044843 - cell cycle G1/S phase transition, GO:1902806 - regulation of cell cycle G1/S phase transition, GO:0008543 - fibroblast growth factor receptor signalling pathway and GO:0072331 - signal transduction by p53 class mediator.

We may thus conclude that our analysis of GBM expression profiles with unsupervised machine learning methods uncovers GBM subtypes which differ in survival and thus an important clinical parameter. The proposed GBM subtypes also differ in several known GBM related biological processes. The technical and biological assessments that were carried out in this thesis allow thus to verify our working hypothesis that integrating data with robust unsupervised modelling has potential to improve our understanding of the molecular mechanisms in GBM. There are however also aspects which demand further research and improvement.

1. Our finding that the majority of genes is significantly differentially expressed across cluster centres is surprising. While all analysis was done carefully and we find plausible interpretations which link our subtypes to known biological consequences of GBM, this finding must be further validated. To rule out that the clustering is confounded with a batch effect in the data, a careful analysis of preprocessing methods is essential.
2. To further deepen our understanding of GBM a second proposition is to consider integrating other data modalities as well. TCGC allows for example to obtain patient genotypes, methylation and copy number modalities, pathology reports and imaging data. To test whether this information provides added value additional analyses should be carried out.

Future work could also seek to apply the methods proposed in this thesis to other diseases where reliable clinical labels are unavailable or should be challenged. Such analysis could for example be applied to other large TCGA projects like COAD (colon cancer) and BRCA (breast cancer).

Appendices to data preprocessing

preprocessing Affymetrix microarray data

First, the phenodata were being imported using the *read.AnnotatedDataFrame()* function, then the file IDs specific to Affymetrix (the .CEL files) were extracted from the phenoData as a list. The function transfers .CEL files to an R object of the Affybatch class. *ReadAffy()* is automatically able to read the different versions of microarray data. Before the read-in process, the data has to be annotated with the phenoData by including them into the *ReadAffy()*. *rma()* is a standard normalisation method of Affymetrix which includes background correction to correct the spatial variation within the arrays. The background correction is being calculated that for each probe, the intensities are positive. Also included is the log transformation to improve the data distribution, quantile normalization for variation correction between arrays and the probe normalization for variation correction within probe sets. This steps serve as the preparation and were repeated for reading Affymetrix data when removing outliers using *arrayQualityMetrics()*.

```
library(affy)
library(arrayQualityMetrics)
library(vsn)

# obtaining phenodata
pData <- read.AnnotatedDataFrame(
"phenodata.csv",
 header=TRUE, row.names=1L, sep="\t")
rownames(phenoData) <- phenoData$file_id_affymetrix

# extracting the .CEL files as a list
celFiles <- paste("",
phenoData$file_name_affymetrix,
sep="", collapse=NULL)

# read and parse the raw data from .CEL files
```

```
affy.data <- ReadAffy(
filenames = celFiles,
sampleNames = sampleNames(phenoData),
phenoData = pData)

# normalisation
affy.prep <- vsnrma(affy.data)
```

The next sections are the actual quality control for Affymetrix microarray, which are used to find the poor quality microarrays. If there are microarrays that re of insufficient quality, they are being removed for the analysis. *arrayQualityMetrics()* provides all the functions and visualisations including box-plots, density plots, heatmap and MA-plot for the quality assessment.

```
# preparing for quality assessment
prepared.data <- prepdata(
expressionset = affy.prep,
intgroup = c(), do.logtransform = TRUE)

# quality control
boxplot.data <- aqm.boxplot(prepared.data)
density.data <- aqm.density(prepared.data)
pca.data <- aqm.pca(prepared.data)
heatmap.data <- aqm.heatmap(prepared.data)
maplot.data <- aqm.maplot(prepared.data)

# outliers
boxplot.out <- boxplot.data@outlier@which
density.out <- density.data@outlier@which
heatmap.out <- heatmap.data@outlier@which
maplot.out <- maplot.data@outlier@which
```

Once the outliers has been generated, the lists were then concatenated into a single list and is being removed from the phenoData and the the preprocessing was being rerun.

```
# obtaining phenodata
pheno.data.outl <- phenoDat[-c(heatmap.out),]

# read and parse the raw data from .CEL files
affy.data.outl <- ReadAffy(
filenames = pheno.data.outl$file_id_affymetrix,
sampleNames = sampleNames(pheno.data.outl),
phenoData = pheno.data.outl)

# normalisation
affy.prep.outl <- vsnrma(affy.data.outl)

# preparing for quality assessment
prepared.data.outl <- prepdata(
expressionset = affy.prep.outl,
intgroup = c(), do.logtransform = TRUE)

# quality control
boxplot.data <- aqm.boxplot(prepared.data.outl)
density.data <- aqm.density(prepared.data.outl)
pca.data <- aqm.pca(prepared.data.outl)
heatmap.data <- aqm.heatmap(prepared.data.outl)
maplot.data <- aqm.maplot(prepared.data.outl)

# save quality assessment in a folder
qm.affy <- list(
    'Boxplot' = boxplot.data$boxplot,
    'Density' = density.data$density,
    'MAPlot' = maplot.data$maplot,
    'Heatmap' = heatmap.data$heatmap,
    'PCAPlot' = pca.data$pca)
aqm.writereport(modules = qm.affy,
    reporttitle = 'QC Report for Affymetrix',
    outdir = "/Users/Desktop/qa_aaffymetrix",
```

```
    arrayTable = pData(pheno.data.outl))
```

preprocessing Agilent microarray data

First, similar to affymetrix, the phenodata were being imported using the *read.AnnotatedDataFrame()* function, then the file names specific to Agilent (the .txt files) were extracted from the phenoData as a list. The function transfers .txt files to an R object class as an expression set (*eset*). *read.maimages()* is automatically able to read and extract the corresponding red (*rProcessedSignal*) and green (*gProcessedSignal*) channels of Agilent microarray data with additional parameter to be chosen. The *read.maimages* function cannot recognise different Agilent version, so different version has to be separately run. *NormalizeVSN()* is a normalisation method of Agilent which includes similar to the *rma()* function a background correction, log transformation, quantile normalization for variation correction between arrays and the probe normalization for variation correction within probe sets. This steps serve as the preparation and were repeated for reading Agilent data, similar to Affymetrix when removing outliers using *arrayQualityMetrics()*.

```
library(limma)
library(arrayQualityMetrics)
library(vsn)

# obtaining phenodata
pheno.data <- read.AnnotatedDataFrame(
    paste(pathName, "phenodataAgilent.csv", sep = ""),
    header=TRUE, row.names=1L, sep="\t")
rownames(pheno.data) <- pheno.data$file_id_agilent

# extracting the agilent files as a list
celFiles <- paste("", pheno.data$file_id_agilent,
    sep="", collapse=NULL)

# read and parse the raw data from agilent .txt files
```

```
agilent.list <- paste("", pheno.data$file_name_agilent,
    sep = "", collapse = NULL)
agi.raw <- read.maimages(
    agilent.list,
    annotation=c(
        "FeatureNum",
        "Sequence",
        "ControlType",
        "ProbeName",
        "GeneName",
        "SystematicName",
        "Description"))
agi.backcorr <- backgroundCorrect(
    agi.raw,
    method = "normexp",
    normexp.method = "rma",
    offset=50)

# normalisation
agi.backcorr$G <- normalizeBetweenArrays(agi.backcorr$G, method="quantile")
agi.backcorr$R <- normalizeBetweenArrays(agi.backcorr$R, method="quantile")

# preparing data for quality control
preparedData <- prepdata(expressionset = backgr.corr, intgroup = c(), do.logtransi

# using heatmap as a quality control
boxplotData <- aqm.boxplot(preparedData)
densityData <- aqm.density(preparedData)
pcaData <- aqm.pca(preparedData)
heatmapData <- aqm.heatmap(preparedData)
maplotData <- aqm.maplot(preparedData)

# outliers
boxplot.out <- boxplot.data@outlier@which
```

```
density.out <- density.data@outlier@which
heatmap.out <- heatmap.data@outlier@which
maplot.out <- maplot.data@outlier@which

pheno.data.outlrm <- pheno.data[-c(heatmap.out),]
agi.raw.outl.rm <- read.maimages(
    pheno.data.outlrm$file_name_agilent,
    annotation=c(
        "FeatureNum",
        "Sequence",
        "ControlType",
        "ProbeName",
        "GeneName",
        "SystematicName",
        "Description"))

# normalisation
agi.vsn <- normalizeVSN(agi.raw.outl.rm) # returning and MAList
agi.eset <- as(agi.vsn, "ExpressionSet")

# quality control
agi.prep.data <- prepdata(
    expressionset = agi.eset,
    intgroup = c(),
    do.logtransform = FALSE)

boxplot.agi <- aqm.boxplot(agi.prep.data)
density.agi <- aqm.density(agi.prep.data)
pca.agi <- aqm.pca(agi.prep.data)
heatmap.agi <- aqm.heatmap(agi.prep.data)
maplot.agi <- aqm.maplot(agi.prep.data)

# save quality assessment in a folder
qm.agi <- list(
```

```
    'Boxplot' = boxplot.agi,
    'Density' = density.agi$density,
    'MAPlot' = maplot.agi$maplot,
    'Heatmap' = heatmap.agi$heatmap,
    'PCAPlot' = pca.agi$pca)
aqm.writereport(
modules = qm.agi,
reporttitle = 'QC Report for Agilent',
outdir = "/Users/Desktop/qa_agilent",
arrayTable = pData(pheno.data.outlrm))
```

The *aqm()* generated results were then stored using the function *aqm.writereport()*, which generates a folder where the results are stored.

preprocessing RNA-Seq data

HTSeq-counts have been preprocessed using *DESeq2* library. *DESeq2* provides a function the can read and parse HTSeq-counts by using the function *DSeqDataSet-FromHTSeqCount()* and convert them into a *DESeqDataSet*. Similar to *vsn()* in Affymetrix and *NormalizeVSN()* in Agilent, *DESeq2* also provides normalization via Variance stabilisation and calibration with the *vst()* function, which returns a normalised *DESeqDataSet*, which is different to the expression set (*eset*). *DESeqDataSet* has to be converted to an expressionist in order to be further applied for the quality control. Subsequently, similar to the quality control of Agilent and Affymetrix, we used the *aqm()* function to generate different quality control measurements.

```
library(DESeq2)
library(arrayQualityMetrics)
library(limma)
library(vsn)

# obtaining phenodata
pheno.data <- read.AnnotatedDataFrame(
    paste(pathName, "MasterThesis/annotation/phenodata.csv", sep = ""),
```

```
    header=TRUE, row.names=1L, sep="\t")
rownames(pheno.data) <- pheno.data$file_id_htseq
filenames.list <- paste("", pheno.data$file_id_htseq, sep = "", collapse = NULL)

# condition
condition.htseq <- paste("", pheno.data$sample_type, sep = "", collapse = NULL)

# target data frame
target.htseq <- data.frame(
    sampleName = filenames.list,
    fileName = filenames.list,
    condition = condition.htseq)

# DESeqDataSet
ddsHTSeq <- DESeqDataSetFromHTSeqCount(
    sampleTable = target.htseq,
    design = ~condition)

# creating a expression set functions
makeExpressionSet <- function(dat, state=colnames(dat)){
  row.names(dat) <- NULL
  dat <- data.matrix(dat)
  pdata <- as.data.frame(state)
  rownames(pdata) <- colnames(dat)

  metadata <- data.frame(labelDescription=c("state"), row.names=c("state"))
  phenoData <- new("AnnotatedDataFrame", data = pdata, varMetadata=metadata)
  dataExp <- ExpressionSet(assayData=dat,  phenoData=phenoData)
  dataExp
}

esetHTSeq <- ExpressionSet(assay(vsdHTSeq)

# automatic quality control using *arrayQualityMetrics()*
```

```
arrayQualityMetrics(esetHTSeq,
    outdir = QC Report for RNA-Seq',
    force = TRUE, do.logtransform = TRUE)
```

Python snippets

For data preparation

After preprocessing the gene expression data, they were saved as a data frame, with an n by m dimension, where n are the genes (rows) and m are the patient samples (columns). The following python snippets below are all the codes that were utilised for preparing the data until the meta-analysis. The function *gene_annotation_GEOparse()* function annotates the probe names with the corresponding gene symbol obtained from the GEOparse library. The *cross_annotate()* function has been used to cross annotate 2 different gene expression data, with the aim of obtaining the same rows and columns between both data, in our case, the cross annotation takes once the probe names has been annotated with the corresponding gene symbol.

```
# -*- coding: utf-8 -*-
#!/usr/bin/env python

import numpy as np
import pandas as pd

def gene_annotation_GEOparse(df, gpl_num = 'GPL570'):
    try:
        import GEOparse
        import re
    except ImportError as error:
        print('GEOparse not installed.
            Try installing GEOparse by either
            "pip install GEOparse" or "conda install GEOparse"')
```

```
    import GEOparse

    # extract annotations from GEO
    gse = GEOparse.get_GEO(geo = gpl_num, destdir = "./")
    GPLAnnot = gse.table

    # set up index of dataframe and GPL
    df = df.loc[df.index.isin(GPLAnnot['ID'])]
    GPLAnnot = GPLAnnot.loc[GPLAnnot['ID'].isin(df.index)]

    if df.shape[0] != GPLAnnot.shape[0]:
        df = df.loc[df.index.isin(GPLAnnot['ID'])]
        GPLAnnot = GPLAnnot.loc[GPLAnnot['ID'].isin(df.index)]
    elif df.shape[0] == GPLAnnot.shape[0]:
        col_sym = [
            a for a in GPLAnnot.columns if re.search('symbol', a.lower())
        ][0]
        df = df.reindex(GPLAnnot['ID'])
        df = df.set_index(GPLAnnot[col_sym]).
                        reset_index().
                        dropna().
                        set_index(col_sym)
    else:
        print(
        'there is something wrong
        with the dataframe.
        Please check for unusual or NaN values.')
    return df


agilent_data = gene_annotation_GEOparse(agilent_data,


def cross_annotate(dataframe, dataframe2, which = 'columns', times = 8):
```

```
    df1 = dataframe.copy()
    df2 = dataframe2.copy()
    for _ in range(times):
        if which == 'columns':
            df1 = df1.loc[:,df1.columns.isin(df2.columns)]
            df2 = df2.loc[:,df2.columns.isin(df1.columns)]
        elif which == 'index':
            df1 = df1.loc[df1.index.isin(df2.index)]
            df2 = df2.loc[df2.index.isin(df1.index)]
        elif which == 'both':
            df1 = df1.loc[:,df1.columns.isin(df2.columns)].loc[df1.index.isin(df2.
            df2 = df2.loc[:,df2.columns.isin(df1.columns)].loc[df2.index.isin(df1.
        else:
            print('Please write the right "which" condition: columns, index or bot

    return df1, df2
```

Clustering analysis and validation

This section displays the python codes for the clustering analysis and validation. These sections provides classes and wrapper functions that include the clustering validation metrics and the clustering algorithms provided for the analysis. The class *Clustering()* summarised all the classes and wrapper functions for the analysis to enable a smooth clustering analysis and validation procedure.

```
# coding: utf-8
import numpy as np
import pandas as pd
import numbers
from sklearn.utils import check_random_state, check_array
import matplotlib.pyplot as plt
import seaborn as sns; sns.set(color_codes=True)
from sklearn import cluster, mixture, metrics
```

```
from scipy.spatial import distance

import math
from sklearn.metrics.pairwise import euclidean_distances

class Metrics(object):
    def __init__(self, X, labels):
        self.labels = np.array(labels)
        self.unqlab = np.unique(self.labels)

        # data
        self.X = np.array(X)

        # number of features/variables
        self.n_features = self.X.shape[1]

        # distance metric
        self.distance = euclidean_distances

        # number of observations
        self.N = len(self.X)    # self.X.shape[0]

        self.length_clusters = {}
        for lab in self.unqlab:
            self.length_clusters[lab] = np.sum(self.labels==lab)

        # number of clusters
        self.n_clusters = len(self.length_clusters) # len(unqlab)

        # center of gravity of clusters
        self.M = np.zeros((self.n_clusters, self.n_features))

        # center of gravity of all points
        self.G = np.zeros((1, self.n_features))
```

```
    for i in range(self.N):
        self.M[self.labels[i]] += self.X[i]
        self.G += self.X[i]

    for i in range(self.n_clusters):
        self.M[i] /= self.length_clusters[i]

    self.G /= self.N

# Total Dispersion Matrix - Total Sum Of Squares
def T(self):
    diff = self.X - self.G
    return diff.T.dot(diff)

# Total Scattering
def TSS(self):
    return np.matrix.trace(self.T())

# Within Group k scatter Matrix
def __WG_k(self, k):
    diff=self.X[self.labels == k] - self.M[self.unqlab == k]
    return diff.T.dot(diff)

# Within Group Matrix
def WG(self):
    within_group = np.zeros((self.n_features, self.n_features))
    for k in self.unqlab:
        within_group += self.__WG_k(k)
    return within_group

# Within k Group Dispersion - Within k Group Sum Of Squares
def __WGSS_k(self, k):
    return np.matrix.trace(self.__WG_k(k))
```

```
# Within Group Dispersion - Within Group Sum Of Squares
def WGSS(self):
    return sum([self.__WGSS_k(k) for k in self.unqlab])

# Between Group Matrix
def BG(self):
    res=np.zeros((self.n_features, self.n_features))
    for k in range(self.n_clusters):
        diff=self.M[k] - self.G
        res = res + diff.T.dot(diff)*self.length_clusters[k]
    return res

# Between Group Dispersion - Between Group Sum Of Squares
def BGSS(self):
    return np.matrix.trace(self.BG())

# Ball and Hall Metrics
def Ball_Hall(self):
    A = 0.0
    for k in self.unqlab:
        B = np.sum(self.distance(self.X[self.labels==k], self.M[k].reshape(1, -1)) *
        A += B / self.length_clusters[k]
    A /= self.n_clusters
    return A

# Banfeld and Raftery Metrics
def Banfeld_Raftery(self):
    A = 0.0
    for k in range(self.n_clusters):
        WGSS_k = self.__WGSS_k(k)
        if WGSS_k < 0.01:
            return 'undefined'
        A += self.length_clusters[k] * np.log(WGSS_k / self.length_clusters[k])
```

```
        return A

    # Calinski and Harabasz Metrics
    def Calinski_Harabasz(self):
        return((self.N - self.n_clusters) * self.BGSS() / ((self.n_clusters -1) *

    # Hartigan Metrics - Log Sum of Squares Ratio
    def Hartigan(self):
        BGSS = self.BGSS()
        WGSS = self.WGSS()
        return np.log((BGSS / WGSS))

    # Friedman and Rubin Determinant Ratio Metrics
    def Friedman_Rubin(self):
        T = self.T()
        WG = self.WG()
        return (np.linalg.det(T)) / (np.linalg.det(WG))

    # log_det_ratio
    def Log_Friedman_Rubin(self):
        return self.N * np.log(self.Friedman_Rubin())

    # Marriot K-squared Determinant Within Metrics
    def Marriot(self):
        WG = self.WG()
        return (self.n_clusters ** 2) * (np.linalg.det(WG))

    # Scott Metrics
    def Scott(self):
        T = self.T()
        WG = self.WG()
        return self.N * np.log((np.linalg.det(T)) / (np.linalg.det(WG)))

    # Ratkowsky and Lance Metrics
```

```
def Ratkowsky_Lance(self):
    A = len(self.BG())
    B = sum([self.BG()[j][j] / self.T()[j][j] for j in range(A)]) / A
    return np.sqrt(B / self.n_clusters)

# Ray and Turi Metrics
def Ray_Turi(self):
    WGSS = self.WGSS()
    alld=self.distance(self.M).reshape(1,-1)[0]
    alld[alld == 0] = np.max(alld)
    Min_d=np.min(alld)
    return WGSS / (self.N * Min_d)

# Trace W Metrics
def Trace_W(self):
    WG = self.WG()
    BG = self.BG()
    return np.matrix.trace((WG.T.dot(BG))) # transpose()

# Xu Metrics
def Xu(self):
    WGSS = self.WGSS()
    return (self.n_features * np.log(np.sqrt((WGSS) /
        ((self.n_features*self.N)**2)) + np.log(self.n_clusters)))

def log_likelihood(self):
    # size of data
    data = self.X
    N, d = data.shape
    # size of clusters
    n = np.bincount(self.labels)
    # number of clusters
    m = len(n)
    # cluster centers
```

```
        M = np.zeros((m,d))
        for i in range(N):
            M[self.labels[i]] += self.X[i]

        for i in range(m):
            M[i] /= n[i]

        centers = [M]
        # compute variance for all clusters
        cl_var = (1.0 / (N - m) / d) * np.sum([np.sum
                    (distance.cdist(self.X[np.where(self.labels == i)],
                    [centers[0][i]], 'euclidean')**2) for i in range(m)])
        # calculate log-likelihood
        log_lh = np.sum([n[i] *
                    np.log10(n[i]) - n[i] *
                        np.log10(N) - ((n[i] * d) / 2) *
                            np.log10(2*np.pi*cl_var) -
                                ((n[i] - 1) * d/ 2) for i in range(m)])
        return log_lh

    def free_parameters(self):
        data = self.X
        N,d = data.shape
        unique_labels = self.unqlab
        K = unique_labels.shape[0]
        r = (K - 1) + (K * d)
        r += 1
        return r

    def BIC(self):
        log_lh = self.log_likelihood()
        penalty = self.free_parameters()
        return log_lh - 0.5 * penalty * np.log(self.N)
```

```
    def AIC(self):
        log_lh = self.log_likelihood()
        penalty = self.free_parameters()
        return log_lh - penalty

def clustering_indices(X, labels, indices = []):
    results = {}

    # Ball and Hall
    if 'BH' in indices:
        results['BH'] = Metrics(X, labels).Ball_Hall()

    # Banfeld and Raftery
    elif 'BR' in indices:
        results['BR'] = Metrics(X, labels).Banfeld_Raftery()

    # Calinski and Harabasz
    elif 'CH' in indices:
        results['CH'] = Metrics(X, labels).Calinski_Harabasz()

    # Hartigan
    elif 'Hartigan' in indices:
        results['Hartigan'] = Metrics(X, labels).Hartigan()

    # Friedman and Rubin
    elif 'FR' in indices:
        results['FR'] = Metrics(X, labels).Friedman_Rubin()

    # Log Friedman and Rubin
    elif 'LFR' in indices:
        results['LFR'] = Metrics(X, labels).Log_Friedman_Rubin()

    # Marriot
    elif 'Marriot' in indices:
```

```
        results['Marriot'] = Metrics(X, labels).Marriot()

    # Scott
    elif 'Scott' in indices:
        results['Scott'] = Metrics(X, labels).Scott()

    # Ratkowsky and Lance
    elif 'RL' in indices:
        results['RL'] = Metrics(X, labels).Ratkowsky_Lance()

    # Ray and Turi
    elif 'RT' in indices:
        results['RT'] = Metrics(X, labels).Ray_Turi()

    # Trace W
    elif 'TW' in indices:
        results['TW'] = Metrics(X, labels).Trace_W()

    # Xu
    elif 'Xu' in indices:
        results['Xu'] = Metrics(X, labels).Xu()

    # Knee point AIC
    elif 'AIC' in indices:
        results['AIC'] = Metrics(X, labels).AIC()

    # Knee point BIC
    elif 'BIC' in indices:
        results['BIC'] = Metrics(X, labels).BIC()

    # Silhouette
    elif 'Silhouette' in indices:
        results['Silhouette'] = metrics.silhouette_score(X, labels)
```

```
    return results

def ClusterWrapper(dataframe, algorithm, nClustMin, nClustMax, kwds):
    scores = []
    silhouette = []
    labels_total = []
    algorithm_name = algorithm.__name__
    for k in range(nClustMin, nClustMax + 1):
        if algorithm_name == 'GaussianMixture'
                or algorithm_name == 'BayesianGaussianMixture':
            labels = algorithm(
                    n_components = k, **kwds).
                    fit(dataframe).
                        predict(dataframe)
        else:
            labels = algorithm(n_clusters = k, **kwds).
                            fit_predict(dataframe)
        labels_total.append(pd.DataFrame({'{}_k{}'.
                    format(algorithm_name, k): list(labels)}))
        #labels_total.append(pd.DataFrame(
                    {algorithm_name + '_k' + str(k):list(labels)}))
    labels_df = pd.concat(labels_total, axis = 1)
    labels_df.index = dataframe.index
    return labels_df

def cluster_wrapper_function(
            clusterTuple,
            dataframe,
            clustMin,
            clustMax):
    cluster_dicts = {}
    for clust in clusterTuple:
        cluster_dicts[clust[0]] = ClusterWrapper(
                    dataframe,
```

```
                    clust[1],
                    clustMin,
                    clustMax,
                    clust[2])
    return cluster_dicts

def metrics_results(
            dataframe,
            labels_dataframe,
            metrics_names):
    """
    This function runs an intern clustering validation from
    the labels with the corresponding dataframe.

    input:
            dataframe:
                    input data from the clustering analysis
            label_dataframe:
                    clustering results from the clustering
                    analysis stored as a dataframe
            metrics_names:
                    list of clustering metrics abbreviations

    output:
            dataframe:
                    clustering validation metrics stored in
                    a dataframe.
    """
    try:
        import cluster_metrics as cm
        import pandas as pd
        import sys
    except ModuleError as error:
        print('Set working directory where cluster_metrics.py is stored.')
```

```
        print('By "import sys" and "sys.path.insert(0,"../python")"')
    scores = []
    for colnames, labels in labels_dataframe.iteritems():
        scores.append(cm.clustering_indices(
                    dataframe,
                    labels,
                    metrics_names))
    return pd.DataFrame(scores)

def stdScaler(metrics, nclustMin):
    """
    This function uses the standard scaler preprocessing
    function from sklearn to transform the metrics dataframe,
    so that all validation metrics scores has the same mean.
    """
    stdDF = pd.DataFrame(StandardScaler().
                    fit_transform(metrics),
                    columns = metrics.columns)
    stdDF.index = stdDF.index + nclustMin
    return stdDF
```

For Differential Expressed Genes

In this section, we implemented the analysis of differential expressed genes with the *limma* package. After obtaining the clustering labels, we used the labels as the design matrix.

```
library(Biobase)
library(GEOquery)
library(illuminaHumanv4.db)
library(vsn)
library(arrayQualityMetrics)
library(limma)
```

```
library(topGO)
library(DOSE)
library(clusterProfiler)
library(org.Hs.eg.db)
library(ggplot2)
library(dplyr)

pathName <- "/Users/marviedemit/Desktop/MasterThesis/Gene_Ontology"
setwd("/Users/marviedemit/Desktop/MasterThesis/Gene_Ontology")
source("PS_et_TM_funcsource.R")

combined.clusterlabels <- read.csv(paste(pathName,
        "/combinedClusterLabels.csv", sep = ""),
        header = TRUE, sep = "\t")
aggregated.clusterlabels <- read.csv(paste(pathName,
        "/aggregatedClusterLabels.csv",
            sep = ""),
                header = TRUE,
                    sep = "\t")

rnaseq <- read.csv(paste(pathName,
        "/rnaseq.csv", sep = ""),
        header = TRUE,
        sep = "\t")
rownames(rnaseq) <- rnaseq$gene_name

affymetrix <- read.csv(paste(pathName,
        "/affymetrix.csv", sep = ""),
        header = TRUE, sep = "\t")
rownames(affymetrix) <- affymetrix$gene_name

aggregated <- read.csv(paste(pathName,
        "/aggregated.csv", sep = ""),
        header = TRUE, sep = "\t")
```

```
rownames(aggregated) <- aggregated$X

gene.names <- as.factor(rownames(rnaseq))
gene.namesDF <- bitr(gene.names,
        fromType="SYMBOL",
        toType = c(
            "ENTREZID",
            "ENSEMBL",
            "GO"), OrgDb="org.Hs.eg.db")
gene.namesBP <- subset(
        gene.namesDF,
        ONTOLOGY == "BP")
idx <- order(gene.namesBP$SYMBOL)
        [!duplicated(sort(gene.namesBP$SYMBOL))]

unq.genes <- gene.namesBP[idx,]
unq.genes.sorted <- unq.genes[
        match(gene.names,
        unq.genes$SYMBOL),]
unq.genes.sorted <- unq.genes.sorted[
            complete.cases(unq.genes.sorted),]

rnaseq <- rnaseq[unq.genes.sorted$SYMBOL,]
affymetrix <- affymetrix[unq.genes.sorted$SYMBOL,]
aggregated <- aggregated[unq.genes.sorted$SYMBOL,]

rnaseq$gene_name <- NULL
affymetrix$gene_name <- NULL
aggregated$X <- NULL

dim(rnaseq)
dim(affymetrix)
dim(aggregated)
```

```
all(rownames(rnaseq) == rownames(affymetrix))
all(rownames(rnaseq) == rownames(aggregated))
# same columns
all(colnames(rnaseq) == colnames(affymetrix))
all(colnames(rnaseq) == colnames(aggregated))

makeExpressionSet <- function(dat, phenoData, featureData){
  # delete the rownames of data or the PGSEA will go wrong.
  dat <- data.matrix(dat)
  dataExp <- ExpressionSet(
        assayData=dat,
        phenoData = phenoData,
        featureData = featureData)
  dataExp
}

fData <- unq.genes.sorted[c("SYMBOL", "ENSEMBL")]
rownames(fData) <- fData$SYMBOL
fData$SYMBOL <- NULL

# checking if the index order of fdata are the same as all
all(rownames(fData) == rownames(rnaseq))
all(rownames(fData) == rownames(affymetrix))
all(rownames(fData) == rownames(aggregated))

featureData <- new("AnnotatedDataFrame",
                   data = fData,
                   varMetadata = data.frame(
                           labelDescription = colnames(fData),
                                   row.names = colnames(fData)))

## arranging phenoData
pData <- read.csv(paste(pathName,
        "/PhenoData.csv", sep = ""),
```

```
        header = TRUE,
        row.names = 1L, sep = "\t")
pData['sample_id_r'] <- make.names(pData$sample_id)
rownames(pData) <- pData$sample_id_r
pData$sample_id <- NULL
pData$sample_id_r <- NULL

# checking if the index order of phenoData are the same as all
all(rownames(pData) == colnames(rnaseq))
all(rownames(pData) == colnames(affymetrix))
all(rownames(pData) == colnames(aggregated))

phenoData <- new("AnnotatedDataFrame",
                 data = pData,
                 varMetadata = data.frame(labelDescription = colnames(pData),
                                          row.names = colnames(pData)))

# creating expressionsets from a dataframe
rnaseq.eset <- makeExpressionSet(
        rnaseq,
        phenoData,
        featureData)
affymetrix.eset <- makeExpressionSet(
        affymetrix,
        phenoData,
        featureData)
aggregated.eset <- makeExpressionSet(
    aggregated,
    phenoData,
    featureData)

#rnaseq.eset <- makeExpressionSet(
        rnaseq[match(rownames(featureData@data),
        rownames(rnaseq)),],
```

```
        phenoData,
        featureData)
#affymetrix.eset <- makeExpressionSet(
        affymetrix[match(rownames(featureData@data),
        rownames(affymetrix)),],
        phenoData,
        featureData)
#aggregated.eset <- makeExpressionSet(
        aggregated[match(rownames(featureData@data),
        rownames(aggregated)),],
        phenoData,
        featureData)

#saveRDS(rnaseq.eset, file="rnaseq_eset.Rds")
#saveRDS(affymetrix.eset, file="affymetrix_eset.Rds")
#saveRDS(aggregated.eset, file="aggregated.eset.Rds")

diffExpress <- function(eset, cluster.labels){
  design <- model.matrix(~factor(cluster.labels))
  fit <- lmFit(eset, design)
  p.fit <- eBayes(fit)
  genelist <- p.fit$F.p.value
  names(genelist) <- rownames(eset@featureData@data)
  genelist
}

selection <- function(allScore){return(allScore < 0.05)}
topfracGeneSel <- function(geneList, topfrac=0.1){
  outList <- rep(FALSE, length(geneList))
  names(outList) <- names(geneList)
  gs <- sort(geneList)
  p.thrs <- gs[ceiling(length(geneList) * topfrac)]
  outList[geneList <= p.thrs] <- TRUE
  return(outList)
```

```
}

allGO2genes <- annFUN.org(
        whichOnto="BP",
        feasibleGenes=NULL,
        mapping="org.Hs.eg.db",
        ID="symbol")

GOanalysis <- function(genelist){
  new("topGOdata",
      ontology="BP",
      allGenes=genelist,
      annot=annFUN.GO2genes,
      GO2genes=allGO2genes,
      geneSel=topfracGeneSel,
      nodeSize=10)
}

### rnaseq
rnaseq.genelist.k.Means <- diffExpress(
            rnaseq.eset,
            combined.clusterlabels$k.Means)

## GO analysis rnaseq kmeans
ngs.km.go <- GOanalysis(rnaseq.genelist.k.Means)
ngs.km.ks <- runTest(ngs.km.go, algorithm="classic", statistic="KS")
pdf('/Users/marviedemit/Desktop/MasterThesis/
    Gene_Ontology/results/GOgraph_rnaseq_kmeans.pdf',
    width = 11,
    height = 9)
showSigOfNodes(
        ngs.km.go,
        score(ngs.km.ks),
        firstSigNodes = 5,
```

```
        useInfo = 'pval')
title(main = 'GO plot of RNA-Seq data from k-Means results',
    cex = 1.5, line = -1.5)
dev.off()

### affymetrix
affymetrix.genelist.k.Means <- diffExpress(
        affymetrix.eset,
        combined.clusterlabels$k.Means)

## GO analysis affymetrix kmeans
affy.km.go <- GOanalysis(affymetrix.genelist.k.Means)
affy.km.ks <- runTest(
    affy.km.go,
    algorithm="classic",
    statistic="ks")
pdf('/Users/marviedemit/Desktop
    /MasterThesis/Gene_Ontology/
    results/GOgraph_affymetrix_kmeans.pdf',
    width = 11,
    height = 9)
showSigOfNodes(
    affy.km.go,
    score(affy.km.ks),
    firstSigNodes = 5, useInfo ='pval')
title(main = 'GO plot of Affymetrix data
    from k-Means GMM results', cex = 1.5, line = -1.5)
dev.off()

### aggregated
aggregated.genelist.k.Means <- diffExpress(
    aggregated.eset,
    aggregated.clusterlabels$k.Means)
```

```
## GO analysis aggregated kmeans
agg.km.go <- GOanalysis(aggregated.genelist.k.Means)
agg.km.ks <- runTest(agg.km.go, algorithm="classic", statistic="ks")
pdf('/Users/marviedemit/Desktop/MasterThesis/
    Gene_Ontology/results/
    GOgraph_aggregated_kmeans.pdf',
    width = 11,
    height = 9)
showSigOfNodes(
    agg.km.go,
    score(agg.km.ks),
    firstSigNodes = 5,
    useInfo ='pval')
title(main = 'GO plot of Aggregated
    data from k-Means results',
    cex = 1.5,
    line = -1.5)
dev.off()
```

For Meta-Analysis

The Meta-analysis code was provided by Dr. Peter Sykacek, that is why only the function I've created without the *fast_pvals* function have been put in. After obtaining the DEG results, with the *p*-values to each genes, we implemented the meta_analysis to combine the *p*-values of RNA-Seq and Affymetrix.

```
# this metaanalysis.py file is provided by Dr. Sykacek
import metaanalysis as ma

def all_exist(avalue, bvalue):
    return all(any(i in j for j in bvalue) for i in avalue)

def meta_analysis(pvalsDF,
```

```
    samples4covdet = 101,
    which = 'fisher'):
if which == 'fisher':
    fisher, _, _, chistats = ma.fast_pvals(
        np.array(pvalsDF),
        samples4covdet = samples4covdet)
    return pd.DataFrame(
        {'fisher':fisher,
        'chi-stats': chistats},
            index = pvalsDF.index)
elif which == 'kost':
    _, kost, _, chistats = ma.fast_pvals(
            np.array(pvalsDF),
            samples4covdet = samples4covdet)
    return pd.DataFrame(
        {'kost':kost,
        'chi-stats': chistats},
            index = pvalsDF.index)
elif which == 'brown':
    _, _, brown, chistats = ma.fast_pvals(
            np.array(pvalsDF),
            samples4covdet = samples4covdet)
    return pd.DataFrame(
        {'brown':brown,
        'chi-stats': chistats},
            index = pvalsDF.index)
elif which == 'all':
    fisher, kost, brown, chistats = ma.fast_pvals(
            np.array(pvalsDF),
            samples4covdet = samples4covdet)
    return pd.DataFrame(
        {'fisher':fisher,
        'kost':kost,
        'brown':brown,
```

```
            'chi-stats': chistats},
                index = pvalsDF.index)
    else:
        print('Choose either "fisher", "kost", "brown" or "all"')

tempath = '/Users/Desktop/MasterThesis
    /data/topGO/diffExpressPvals'

for i in diffExpressPvals:
    globals()[i.replace('.csv', '')] = loadExpressPvals(
            i, tempath, i.replace('_', ' ').replace('.csv', '').split()[-1])

# concatenate the p-values
affyGOCombPvals = pd.concat(
    [affymetrix_genelist_kmeans,
    affymetrix_genelist_kmgm,
    affymetrix_genelist_spec], axis = 1)
rnaseqGOCombPvals = pd.concat(
    [rnaseq_genelist_kmeans,
    rnaseq_genelist_kmgm,
    rnaseq_genelist_spec], axis = 1)
aggGOCombPvals = pd.concat(
    [aggregated_genelist_kmeans,
    aggregated_genelist_kmgm,
    aggregated_genelist_spec], axis = 1)

# meta analysis
aggGOMeta = meta_analysis(aggGOCombPvals, 1000, 'fisher')
#aggGOMeta = aggGOMeta[(aggGOMeta['fisher'] != 0)]

affyGOMeta = meta_analysis(affyGOCombPvals, 1000, 'fisher')
rnaseqGOMeta = meta_analysis(rnaseqGOCombPvals, 1000, 'fisher')
# meta analysis by aggregating the results from each clustering algorithm
rnaseq_affy_kmeans = meta_analysis(
```

```
    pd.concat(
        [rnaseqGOCombPvals['kmeans'],
        affyGOCombPvals['kmeans']], axis = 1).
            dropna(how = 'any'), 1000, 'fisher')

# combined RNA-Seq and Affymetrix
rnaseq_affy_comb = meta_analysis(
    pd.concat([rnaseq_affy_kmeans['fisher'],
    rnaseq_affy_spec['fisher'],
    rnaseq_affy_kmgm['fisher']], axis = 1), 1000, 'fisher')
```

Gene Set Enrichment Analysis (GSEA)

We implemented the *clusterProfiler* library to generate the GSEA analysis results. First we prepared the gene list with the p-values annotated with each corresponding genes. This gene list is then the input for the *enrichGO()* which return the enrichment Gene Ontology categories after FDR-BH control.

```
library(limma)
library(topGO)
library(DOSE)
library(clusterProfiler)

## set-up working directory
pathName <- "/Users/Desktop/MasterThesis/data/topGO/metapvalsGO/"
#dirName <- "/data/topGo"
setwd("/Users/Desktop/MasterThesis/data/topGO/metapvalsGO/")
#source("PS_et_TM_funcsource.R")
# source("./ebm_modified.R")

rnaseq.meta <- read.csv(paste(pathName, "/rnaseqMETA.csv", sep = ""), header = TRU
affy.meta <- read.csv(paste(pathName, "/affyMETA.csv", sep = ""), header = TRUE, s
aggregated.meta <- read.csv(paste(pathName, "/aggMETA.csv", sep = ""), header = TR
rnaseq.affy.meta <- read.csv(paste(pathName, "/rnaseqaffyMETA.csv", sep = ""), hea
```

```r
# creating gene lists
rnaseq.meta.genelist <- rnaseq.meta$fisher
names(rnaseq.meta.genelist) <- rnaseq.meta$gene_symbol

affy.meta.genelist <- affy.meta$fisher
names(affy.meta.genelist) <- affy.meta$gene_symbol

agg.meta.genelist <- aggregated.meta$fisher
names(agg.meta.genelist) <- aggregated.meta$gene_symbol

rnaseq.affy.meta.genelist <- rnaseq.affy.meta$fisher
names(rnaseq.affy.meta.genelist) <- rnaseq.affy.meta$gene_symbol

# GO analysis Biological Process
setwd()

# RNA-Seq
rnaseq.meta.genes <- names(rnaseq.meta.genelist)[abs(rnaseq.meta.genelist) != 0]
rnaseq.meta.genes.df <- bitr(rnaseq.meta.genes, fromType = "SYMBOL",
                             toType = c("ENSEMBL", "ENTREZID", 'GO'),
                              OrgDb = org.Hs.eg.db)

rnaseq.meta.genes.ego <- enrichGO(
  gene          = rnaseq.meta.genes.df$ENSEMBL,
  keyType       = "ENSEMBL",
  OrgDb         = org.Hs.eg.db,
  ont           = "BP",
  pAdjustMethod = "BH",
  pvalueCutoff  = 0.01,
  qvalueCutoff  = 0.05,
  readable      = TRUE
  )
```

```r
# affymetrix
affy.meta.genes <- names(affy.meta.genelist)[abs(affy.meta.genelist) != 0]
affy.meta.genes.df <- bitr(
  affy.meta.genes, fromType = "SYMBOL",
  toType = c("ENSEMBL", "ENTREZID", 'GO'),
  OrgDb = org.Hs.eg.db
)

affy.meta.genes.ego <- enrichGO(
  gene          = affy.meta.genes.df$ENSEMBL,
  keyType       = "ENSEMBL",
  OrgDb         = org.Hs.eg.db,
  ont           = "BP",
  pAdjustMethod = "BH",
  pvalueCutoff  = 0.01,
  qvalueCutoff  = 0.05,
  readable      = TRUE
)

# aggregated
agg.meta.genes <- names(agg.meta.genelist)[abs(agg.meta.genelist) != 0]
agg.meta.genes.df <- bitr(
  agg.meta.genes, fromType = "SYMBOL",
  toType = c("ENSEMBL", "ENTREZID", 'GO'),
  OrgDb = org.Hs.eg.db
)
agg.meta.genes.ego <- enrichGO(
  gene          = agg.meta.genes.df$ENSEMBL,
  keyType       = "ENSEMBL",
  OrgDb         = org.Hs.eg.db,
  ont           = "BP",
  pAdjustMethod = "BH",
  pvalueCutoff  = 0.01,
```

```
  qvalueCutoff  = 0.05,
  readable      = TRUE
)

# combined
rnaseq.affy.meta.genes <- names(
    rnaseq.affy.meta.genelist)[abs(rnaseq.affy.meta.genelist) != 0]
rnaseq.affy.meta.genes.df <- bitr(
  rnaseq.affy.meta.genes, fromType = "SYMBOL",
  toType = c("ENSEMBL", "ENTREZID", 'GO'),
  OrgDb = org.Hs.eg.db
)
rnaseq.affy.meta.genes.ego <- enrichGO(
  gene          = rnaseq.affy.meta.genes.df$ENSEMBL,
  keyType       = "ENSEMBL",
  OrgDb         = org.Hs.eg.db,
  ont           = "BP",
  pAdjustMethod = "BH",
  pvalueCutoff  = 0.01,
  qvalueCutoff  = 0.05,
  readable      = TRUE
)
```

References

Abdi H, Williams LJ. 2010. Principal component analysis. *Wiley interdisciplinary reviews: computational statistics* **2**:433–459.

Ackermann MR, Blömer J, Sohler C. 2010. Clustering for metric and nonmetric distance measures. *ACM Transactions on Algorithms (TALG)* **6**:59.

Adamson C, Kanu OO, Mehta AI, Di C, Lin N, Mattox AK, Bigner DD. 2009. Glioblastoma multiforme: A review of where we have been and where we are going. *Expert opinion on investigational drugs* **18**:1061–1083.

Aggarwal CC, Reddy CK. 2013. Data clustering: Algorithms and applications. CRC press.

Akaike H. 2011. Akaike's information criterion. In:. *International encyclopedia of statistical science*. Springer, pp. 25–25.

Alexa A, Rahnenfuhrer J. 2010. TopGO: Enrichment analysis for gene ontology. *R package version* **2**:2010.

Alexa A, Rahnenführer J. 2009. Gene set enrichment analysis with topGO. *Bioconductor Improv* **27**.

Allison DB, Cui X, Page GP, Sabripour M. 2006. Microarray data analysis: From disarray to consolidation and consensus. *Nature reviews genetics* **7**:55.

Anders S, Huber W. 2010. Differential expression analysis for sequence count data. *Genome biology* **11**:R106.

Association ABT, others. 2016. Brain tumor statistics. *Available at:. Accessed May*

2.

Azevedo-Filho A, Shachter RD. 1994. Laplace's method approximations for probabilistic inference in belief networks with continuous variables. In:. *Uncertainty proceedings 1994*. Elsevier, pp. 28–36.

Banfield JD, Raftery AE. 1993. Model-based gaussian and non-gaussian clustering. *Biometrics*:803–821.

Beebe NH. 2018. A bibliography of publications about the python scripting and programming language.

Berkhin P. 2006. A survey of clustering data mining techniques. In:. *Grouping multidimensional data*. Springer, pp. 25–71.

Bielza C, Larrañaga P. 2004. Bayesian information criterion (bic). *Dictionary of Bioinformatics and Computational Biology*.

Bishop CM. 2012. Pattern recognition and machine learning, 2006 **60**:78–78.

Brat DJ, Van Meir EG. 2004. Vaso-occlusive and prothrombotic mechanisms associated with tumor hypoxia, necrosis, and accelerated growth in glioblastoma. *Laboratory investigation* **84**:397.

Buse A. 1982. The likelihood ratio, wald, and lagrange multiplier tests: An expository note. *The American Statistician* **36**:153–157.

Caliński T, Harabasz J. 1974. A dendrite method for cluster analysis. *Communications in Statistics-theory and Methods* **3**:1–27.

Crocetti E, Trama A, Stiller C, Caldarella A, Soffietti R, Jaal J, Weber DC, Ricardi U, Slowinski J, Brandes A, others. 2012. Epidemiology of glial and non-glial brain tumours in europe. *European journal of cancer* **48**:1532–1542.

Davidson-Pilon C, Kalderstam J, Zivich P, Kuhn B, Fiore-Gartland A, Moneda L, Gabriel, WIlson D, Parij A, Stark K, Anton S, Besson L, Jona, Gadgil H, Golland D, Hussey S, Kumar R, Noorbakhsh J, Klintberg A, Kaluzka J, Slavitt I, Martin E, Ochoa E, Albrecht D, dhuynh, Zgonjanin D, Chen D, Fournier C, Arturo, Rendeiro AF. 2019. CamDavidsonPilon/lifelines: V0.21.1. https://doi.org/10.5281/zenodo.

2652543.

Dempster AP, Laird NM, Rubin DB. 1977. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society. Series B (methodological)*:1–38.

DeRisi JL, Iyer VR, Brown PO. 1997. Exploring the metabolic and genetic control of gene expression on a genomic scale. *Science* **278**:680–686.

Desgraupes B. 2013. ClusterCrit: Clustering indices. *R package version* **1**:4–5.

Donath W, Hoffman A. 1972. Algorithms for partitioning of graphs and computer logic based on eigenvectors of connections matrices. *IBM Technical Disclosure Bulletin* **15**.

Drăghici S. 2016. Statistics and data analysis for microarrays using r and bioconductor. Chapman; Hall/CRC.

Dunn GP, Andronesi OC, Cahill DP. 2013. From genomics to the clinic: Biological and translational insights of mutant idh1/2 in glioma. *Neurosurgical focus* **34**:E2.

Dunwoodie LJ, Poehlman WL, Ficklin SP, Feltus FA. 2018. Discovery and validation of a glioblastoma co-expressed gene module. *Oncotarget* **9**:10995.

Eisen MB, Spellman PT, Brown PO, Botstein D. 1998. Cluster analysis and display of genome-wide expression patterns. *Proceedings of the National Academy of Sciences* **95**:14863–14868.

Elston R. 1991. On fisher's method of combining p-values. *Biometrical journal* **33**:339–345.

Estivill-Castro V. 2002. Why so many clustering algorithms: A position paper. *ACM SIGKDD explorations newsletter* **4**:65–75.

Fiedler M. 1975. A property of eigenvectors of nonnegative symmetric matrices and its application to graph theory. *Czechoslovak Mathematical Journal* **25**:619–633.

Fisher RA. 2006. Statistical methods for research workers. Genesis Publishing Pvt Ltd.

Friedman HP, Rubin J. 1967. On some invariant criteria for grouping data. *Journal*

of the American Statistical Association **62**:1159–1178.

Gentleman RC, Carey VJ, Bates DM, Bolstad B, Dettling M, Dudoit S, Ellis B, Gautier L, Ge Y, Gentry J, others. 2004. Bioconductor: Open software development for computational biology and bioinformatics. *Genome biology* **5**:R80.

Gentleman R, Carey V, Huber W, Irizarry R, Dudoit S. 2006. Bioinformatics and computational biology solutions using r and bioconductor. Springer Science & Business Media.

Halkidi M, Batistakis Y, Vazirgiannis M. 2001. On clustering validation techniques. *Journal of intelligent information systems* **17**:107–145.

Hartigan JA, Wong MA. 1979. Algorithm as 136: A k-means clustering algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)* **28**:100–108.

Heller MJ. 2002. DNA microarray technology: Devices, systems, and applications. *Annual review of biomedical engineering* **4**:129–153.

Henriksen M, Johnsen KB, Andersen HH, Pilgaard L, Duroux M. 2014. MicroRNA expression signatures determine prognosis and survival in glioblastoma multiforme—a systematic overview. *Molecular neurobiology* **50**:896–913.

Hunter JD. 2007. Matplotlib: A 2D graphics environment. *Computing in science & engineering* **9**:90–95.

Irizarry RA, Gautier L, Bolstad BM, Miller C. 2006. Methods for affymetrix oligonucleotide arrays. *R package version 1.12* **1**.

Jasra A, Holmes CC, Stephens DA. 2005. Markov chain monte carlo methods and the label switching problem in bayesian mixture modeling. *Statistical Science*:50–67.

Jemal A, Siegel R, Ward E, Hao Y, Xu J, Thun MJ. 2009. Cancer statistics, 2009. *CA: a cancer journal for clinicians* **59**:225–249.

Jones E, Oliphant T, Peterson P. 2014. {SciPy}: Open source scientific tools for {python}.

Joyce JM. 2011. Kullback-leibler divergence. In:. *International encyclopedia of statistical science*. Springer, pp. 720–722.

Kaever A, Landesfeind M, Feussner K, Morgenstern B, Feussner I, Meinicke P. 2014.

Meta-analysis of pathway enrichment: Combining independent and dependent omics data sets. *PLoS One* **9**:e89297.

Kanu OO, Hughes B, Di C, Lin N, Fu J, Bigner DD, Yan H, Adamson C. 2009. Glioblastoma multiforme oncogenomics and signaling pathways. *Clinical medicine. Oncology* **3**:CMO–S1008.

Kaplan EL, Meier P. 1958. Nonparametric estimation from incomplete observations. *Journal of the American statistical association* **53**:457–481.

Kircher M, Stenzel U, Kelso J. 2009. Improved base calling for the illumina genome analyzer using machine learning strategies. *Genome biology* **10**:R83.

Kogenaru S, Yan Q, Guo Y, Wang N. 2012. RNA-seq and microarray complement each other in transcriptome profiling. *BMC genomics* **13**:629.

Kost JT, McDermott MP. 2002. Combining dependent p-values. *Statistics & Probability Letters* **60**:183–190.

Krzanowski WJ, Lai Y. 1988. A criterion for determining the number of groups in a data set using sum-of-squares clustering. *Biometrics*:23–34.

Kunkle BW, Yoo C, Roy D. 2013. Reverse engineering of modified genes by bayesian network analysis defines molecular determinants critical to the development of glioblastoma. *PloS one* **8**:e64140.

Libbrecht MW, Noble WS. 2015. Machine learning applications in genetics and genomics. *Nature Reviews Genetics* **16**:321.

Liu Y, Li Z, Xiong H, Gao X, Wu J. 2010. Understanding of internal clustering validation measures. In:. *Data mining (icdm), 2010 ieee 10th international conference on*. IEEE, pp. 911–916.

Livak KJ, Schmittgen TD. 2001. Analysis of relative gene expression data using real-time quantitative pcr and the 2- $\Delta\Delta$ct method. *methods* **25**:402–408.

Maaten L van der, Hinton G. 2008. Visualizing data using t-sne. *Journal of machine learning research* **9**:2579–2605.

Marriott F. 1971. Practical problems in a method of cluster analysis. *Biometrics*:501–

514.

Masui K, Cloughesy T, Mischel P. 2012. Molecular pathology in adult high-grade gliomas: From molecular diagnostics to target therapies. *Neuropathology and applied neurobiology* **38**:271–291.

McKinney W. 2015. Pandas: A python data analysis library. *see http://pandas. pydata. org/. Google Scholar*.

McLachlan G, Krishnan T. 2007. The em algorithm and extensions. John Wiley & Sons. Vol. 382.

Meliso FM, Hubert CG, Galante PAF, Penalva LO. 2017. RNA processing as an alternative route to attack glioblastoma. *Human genetics* **136**:1129–1141.

Miller MB, Tang Y-W. 2009. Basic concepts of microarrays and potential applications in clinical microbiology. *Clinical microbiology reviews* **22**:611–633.

Milligan GW, Cooper MC. 1985. An examination of procedures for determining the number of clusters in a data set. *Psychometrika* **50**:159–179.

Minka T. 1998. Expectation-maximization as lower bound maximization. Tutorial published on the web at http://research. microsoft. com/users/minka/papers/minka-em-tut. ps. gz.

Mooney M, Bond J, Monks N, Eugster E, Cherba D, Berlinski P, Kamerling S, Marotti K, Simpson H, Rusk T, others. 2013. Comparative rna-seq and microarray analysis of gene expression changes in b-cell lymphomas of canis familiaris. *PloS one* **8**:e61088.

Morozova O, Hirst M, Marra MA. 2009. Applications of new sequencing technologies for transcriptome analysis. *Annual review of genomics and human genetics* **10**:135–151.

Murat A, Migliavacca E, Gorlia T, Lambiv WL, Shay T, Hamou M-F, De Tribolet N, Regli L, Wick W, Kouwenhoven MC, others. 2008. Stem cell–related "self-renewal" signature and high epidermal growth factor receptor expression associated with resistance to concomitant chemoradiotherapy in glioblastoma. *Journal of clinical oncology* **26**:3015–3024.

Nakada M, Nakada S, Demuth T, Tran N, Hoelzinger D, Berens M. 2007. Molecular

targets of glioma invasion. *Cellular and molecular life sciences* **64**:458.

Nakada M, Kita D, Watanabe T, Hayashi Y, Teng L, Pyko IV, Hamada J-I. 2011. Aberrant signaling pathways in glioma. *Cancers* **3**:3242–3278.

Nasrabadi NM. 2007. Pattern recognition and machine learning. *Journal of electronic imaging* **16**:049901.

Neal RM, Hinton GE. 1998. A view of the em algorithm that justifies incremental, sparse, and other variants. In:. *Learning in graphical models*. Springer, pp. 355–368.

Network CGAR, others. 2017a. GDC api user's guide. *Genomic Data Commons* **1**:213.

Network CGAR, others. 2017b. GDC data access policy. *Genomic Data Commons* **1**:213.

Nikiforova MN, Hamilton RL. 2011. Molecular diagnostics of gliomas. *Archives of pathology & laboratory medicine* **135**:558–568.

Ohgaki H, Kleihues P. 2012. The definition of primary and secondary glioblastoma. *Clinical cancer research*:clincanres–3002.

Pagliarulo V, Datar RH, Cote RJ. 2002. Role of genetic and expression profiling in pharmacogenomics: The changing face of patient management. *Current issues in molecular biology* **4**:101–110.

Pan J-X, Fang K-T. 2002. Maximum likelihood estimation. In:. *Growth curve models and statistical diagnostics*. Springer, pp. 77–158.

Papastamoulis P. 2015. Label. switching: An r package for dealing with the label switching problem in mcmc outputs. *arXiv preprint arXiv:1503.02271*.

Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, others. 2011. Scikit-learn: Machine learning in python. *Journal of machine learning research* **12**:2825–2830.

Phillips HS, Kharbanda S, Chen R, Forrest WF, Soriano RH, Wu TD, Misra A, Nigro JM, Colman H, Soroceanu L, others. 2006. Molecular subclasses of high-grade glioma predict prognosis, delineate a pattern of disease progression, and resemble

stages in neurogenesis. *Cancer cell* **9**:157–173.

Poole W, Gibbs DL, Shmulevich I, Bernard B, Knijnenburg TA. 2016. Combining dependent p-values with an empirical adaptation of brown's method. *Bioinformatics* **32**:i430–i436.

Purkait S, Mallick S, Sharma V, Kumar A, Pathak P, Jha P, Biswas A, Julka PK, Gupta D, Suri A, others. 2016. A simplified approach for molecular classification of glioblastomas (gbms): Experience from a tertiary care center in india. *Brain tumor pathology* **33**:183–190.

R Development Core Team. 2008. R: A language and environment for statistical computing. http://www.R-project.org.

Ratkowsky D, Lance G. 1978. Criterion for determining the number of groups in a classification.

Ray S, Turi RH. 1999. Determination of number of clusters in k-means clustering and application in colour image segmentation. In:. *Proceedings of the 4th international conference on advances in pattern recognition and digital techniques*. Calcutta, India, pp. 137–143.

Reynolds D. 2015. Gaussian mixture models. *Encyclopedia of biometrics*:827–832.

Richardson M. 2009. Principal component analysis. *URL: http://people. maths. ox. ac. uk/richardsonm/SignalProcPCA. pdf (last access: 3.5. 2013). Aleš Hladnik Dr., Ass. Prof., Chair of Information and Graphic Arts Technology, Faculty of Natural Sciences and Engineering, University of Ljubljana, Slovenia ales. hladnik@ ntf. uni-lj. si* **6**:16.

Rokach L, Maimon O. 2005. Clustering methods. In:. *Data mining and knowledge discovery handbook*. Springer, pp. 321–352.

Rong Y, Durden DL, Van Meir EG, Brat DJ. 2006. 'Pseudopalisading'necrosis in glioblastoma: A familiar morphologic feature that links vascular pathology, hypoxia, and angiogenesis. *Journal of Neuropathology & Experimental Neurology* **65**:529–539.

Schena M. 2003. Microarray analysis. Wiley-Liss Hoboken, NJ.

Schena M, Shalon D, Davis RW, Brown PO. 1995. Quantitative monitoring of gene

expression patterns with a complementary dna microarray. *Science* **270**:467–470.

Schulze A, Downward J. 2001. Navigating gene expression using microarrays—a technology review. *Nature cell biology* **3**:E190.

Simon RM, Korn EL, McShane LM, Radmacher MD, Wright GW, Zhao Y. 2003. Design and analysis of dna microarray investigations. Springer Science & Business Media.

Smyth GK. 2005. Limma: Linear models for microarray data. In:. *Bioinformatics and computational biology solutions using r and bioconductor*. Springer, pp. 397–420.

Soni N, Ganatra A. 2012. Categorization of several clustering algorithms from different perspective: A review. *International Journal of*.

Stephens M. 2000. Dealing with label switching in mixture models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **62**:795–809.

Sturn A, Quackenbush J, Trajanoski Z. 2002. Genesis: Cluster analysis of microarray data. *Bioinformatics* **18**:207–208.

Tan P-N, Steinbach M, Kumar V. 2005. Association analysis: Basic concepts and algorithms. *Introduction to Data mining*:327–414.

Tarazona S, García-Alcalde F, Dopazo J, Ferrer A, Conesa A. 2011. Differential expression in rna-seq: A matter of depth. *Genome research*:gr–124321.

Tomczak K, Czerwińska P, Wiznerowicz M. 2015. The cancer genome atlas (tcga): An immeasurable source of knowledge. *Contemporary oncology* **19**:A68.

Van't Veer LJ, Dai H, Van De Vijver MJ, He YD, Hart AA, Mao M, Peterse HL, Van Der Kooy K, Marton MJ, Witteveen AT, others. 2002. Gene expression profiling predicts clinical outcome of breast cancer. *nature* **415**:530.

Verhaak RG, Hoadley KA, Purdom E, Wang V, Qi Y, Wilkerson MD, Miller CR, Ding L, Golub T, Mesirov JP, others. 2010. Integrated genomic analysis identifies clinically relevant subtypes of glioblastoma characterized by abnormalities in pdgfra, idh1, egfr, and nf1. *Cancer cell* **17**:98–110.

Von Luxburg U. 2007. A tutorial on spectral clustering. *Statistics and computing*

17:395–416.

Von Neubeck C, Seidlitz A, Kitzler H, Beuthien-Baumann B, Krause M. 2015. Glioblastoma multiforme: Emerging treatments and stratification markers beyond new drugs. *The British journal of radiology* **88**:20150354.

Wallner KE, Galicich JH, Krol G, Arbit E, Malkin MG. 1989. Patterns of failure following treatment for glioblastoma multiforme and anaplastic astrocytoma. *International Journal of Radiation Oncology* Biology* Physics* **16**:1405–1409.

Walt S van der, Colbert SC, Varoquaux G. 2011. The numpy array: A structure for efficient numerical computation. *Computing in Science & Engineering* **13**:22–30.

Wang A, Zhang G. 2017. Differential gene expression analysis in glioblastoma cells and normal human brain cells based on geo database. *Oncology letters* **14**:6040–6044.

Wang Z, Gerstein M, Snyder M. 2009. RNA-seq: A revolutionary tool for transcriptomics. *Nature reviews genetics* **10**:57.

Weller M, Bent M van den, Hopkins K, Tonn JC, Stupp R, Falini A, Cohen-Jonathan-Moyal E, Frappaz D, Henriksson R, Balana C, others. 2014. EANO guideline for the diagnosis and treatment of anaplastic gliomas and glioblastoma. *The lancet oncology* **15**:e395–e403.

Wen PY, Kesari S. 2008. Malignant gliomas in adults. *New England Journal of Medicine* **359**:492–507.

Willenbrock H, Salomon J, Søkilde R, Barken KB, Hansen TN, Nielsen FC, Møller S, Litman T. 2009. Quantitative miRNA expression analysis: Comparing microarrays with next-generation sequencing. *Rna* **15**:2028–2034.

Wold S, Esbensen K, Geladi P. 1987. Principal component analysis. *Chemometrics and intelligent laboratory systems* **2**:37–52.

Wolpert DH, Macready WG. 1997. No free lunch theorems for optimization. *IEEE transactions on evolutionary computation* **1**:67–82.

Xiang Y, Zhang C-Q, Huang K. 2012. Predicting glioblastoma prognosis networks using weighted gene co-expression network analysis on tcga data. In:. *BMC bioin-*

formatics. BioMed Central. 2, Vol. 13, p. S12.

Xu P, Yang J, Liu J, Yang X, Liao J, Yuan F, Xu Y, Liu B, Chen Q. 2018. Identification of glioblastoma gene prognosis modules based on weighted gene co-expression network analysis. *BMC medical genomics* **11**:96.

Xu R, Wunsch D. 2005. Survey of clustering algorithms. *IEEE Transactions on neural networks* **16**:645–678.

Yang S, Gao K, Li W. 2019. Identification of hub genes and pathways in glioblastoma by bioinformatics analysis. *Oncology letters* **17**:1035–1041.

Yeo GW. 2016. RNA processing: Disease and genome-wide probing. Springer. Vol. 907.

Zhao Q, Hautamaki V, Fränti P. 2008. Knee point detection in bic for detecting the number of clusters. In:. *International conference on advanced concepts for intelligent vision systems*. Springer, pp. 664–673.

Zhao S, Fung-Leung W-P, Bittner A, Ngo K, Liu X. 2014. Comparison of rna-seq and microarray in transcriptome profiling of activated t cells. *PloS one* **9**:e78644.

Zhu Y, Parada LF. 2002. The molecular and genetic basis of neurological tumours. *Nature Reviews Cancer* **2**:616.