Sample Size Evaluation and Comparison of K-Means Clusterings of RNA-Seq Gene Expression Data

A Thesis Submitted to the
College of Graduate and Postdoctoral Studies
in Partial Fulfillment of the Requirements
for the degree of Master of Science
in the Department of Computer Science
University of Saskatchewan
Saskatoon

By
H M Zabir Haque

Permission to Use

Head of the Department of Computer Science
176 Thorvaldson Building
110 Science Place
University of Saskatchewan
Saskatoon, Saskatchewan
Canada
S7N 5C9

Or

Dean
College of Graduate and Postdoctoral Studies
University of Saskatchewan
116 Thorvaldson Building, 110 Science Place
Saskatoon, Saskatchewan S7N 5C9
Canada

Abstract

The process by which DNA is transformed into gene products, such as RNA and proteins, is called gene expression. Gene expression profiling quantifies the expression of genes (amount of RNA) in a particular tissue at a particular time. Two commonly used high-throughput techniques for gene expression analysis are DNA microarrays and RNA-Seq, with RNA-Seq being the newer technique based on high-throughput sequencing.

Statistical analysis is needed to deal with complex datasets — one commonly used statistical tool is clustering. Clustering comparison is an existing area dedicated to comparing multiple clusterings from one or more clustering algorithms. However, there has been limited application of cluster comparisons to clusterings of RNA-Seq gene expression data. In particular, cluster comparisons are useful in order to test the differences between clusterings obtained using a single algorithm when using different samples for clustering.

Here we use a metric for cluster comparisons that is a variation of existing metrics. The metric is simply the minimal number of genes that need to be moved from one cluster to another in one given clustering to produce another given clustering. As the metric only has genes (or elements) as units, it is easy to interpret for RNA-Seq analysis. Moreover, three different algorithmic techniques — brute force, branch-and-bound, and maximal bipartite matching — for computing the proposed metric exactly are compared in terms of time to compute, with bipartite matching being significantly more time efficient.

This metric is then applied to the important issue of understanding the effect of increasing the number of RNA-Seq samples to clusterings. Three datasets were used where a large number of samples were available: mouse embryonic stem cell tissue data, *Drosophila melanogaster* data from multiple tissues and micro-climates, and a mouse multi-tissue dataset. For each, a reference clustering was computed from all of the samples, and then it was compared to clusterings created from smaller subsets of the samples. All clusterings were created using a standard heuristic K-means clustering algorithm, while also systematically varying the numbers of clusters, and also using both Euclidean distance and Manhattan distance. The clustering comparisons suggest that for the three large datasets tested, there seems to be a limited impact of adding more RNA-Seq samples on K-means clusterings using both Euclidean distance and Manhattan distance (Manhattan distance gives a higher variation) beyond some small number of samples. That is, the clusterings compiled based on a limited number of samples were all either quite similar to the reference clustering or did not improve as additional samples were added. These findings were the same for different numbers of clusters. The methods developed could also be applied to other clustering comparison problems.

Acknowledgements

First, I would like to thank the almighty — Allah for giving me the chance to initiate this research and be able me to complete the work successfully. I always feel his blessing at every step of my life. Allah has embraced us with his grace. The blessing of the almighty helps me to keep patient and work continuously.

I would be glad to express my gratitude to my supervisor, Dr. Ian McQuillan, for his support, and assistance throughout my Master's program. Ian's technical and editorial advice was essential to the completion of this dissertation. His meticulous proofreading allowed this thesis to move in the right direction. As a student of engineering, I never learned the field of biology. Ian's course and supervision helped me to become much more comfortable in understanding the biological parts of Bioinformatics. Besides being a supervisor, I have always found him to be a friend. The door to Ian's office was always open whenever I faced any problem or had a question about my research or writing.

I thank my committee members: Dr. Tony Kusalik, Dr. Michael Horsch, and David Schneider for their valuable comments and feedback on the research. I would like to give special thanks to Dr. Kusalik for his Readings in Bioinformatics course. In this course, I had the opportunity to learn many aspects of bioinformatics. I would also like to thank Dr. Kevin Stanley for his machine learning courses. Before taking Dr. Stanley's courses, I hardly knew the area of machine learning, but he helped me to link all the ideas of machine learning in a straight line which helped me to visualize the broader view of the area, and its diverse application areas.

I am grateful to all of my lab members and friends who helped me a lot. I thank Katie Ovens specifically for helping with datasets. Special mention goes to Gwen Lancaster, Sophie Findlay, Daniel Hogan, Md. Sowgat Ibne Mahmud, Faheem Abrar, Najeeb Ullah Khan, Musharaf Mamun, Kazi Ashik Ahmed, Nazifa Khan, Rifat Zahan, Farhad Maleki, and Kimberly MacKay.

Finally, I want to convey a very intense acknowledgment to my parents: Dr. Md. Zahurul Haque and Helen Haque and my relatives for providing me with endless support and constant inspiration. Without them, it would have been hard for me to accomplish the research work.

Last but the not least, I also like to show my appreciation to the University of Saskatchewan, and to Ian for their financial assistance.

Contents

List of Tables

List of Figures

List of Abbreviations

CCD	Cluster Comparison Distance
CD	Clustering Distance
CE	Classification Error
DBI	Davies-Bouldin Index
DI	Dunn Index
DNA	Deoxyribonucleic Acid
DSC	Dice Similarity Coefficient
FMS	Fowlkes-Mallows Score
FPKM	Fragments Per Kilobase Million
GEO	Gene Expression Omnibus
GFF	General Feature Format
GTF	Gene Transfer Format
KL Divergence	Kullback-Leibler Divergence
PCC	Pearson Correlation Coefficient
RNA	Ribonucleic Acid
RNA-Seq	RNA Sequencing
SNP	Single Nucleotide Polymorphism
STAR	Spliced Transcripts Alignment to a Reference

1 Introduction

Gene expression profiling is a method to identify the activity of genes within cells at a given moment, tissue, or condition [66]. Using the activity of genes, gene expression profiling can help draw conclusions about cell type, state, environment, or biological processes. Gene expression analysis is especially useful for disease diagnosis or drug development. For example, it can help with determining the toxicity of a drug, or the treatment of cancer [79]. It is therefore one of the most important tasks for answering biological questions.

Many modern technologies for expression profiling are high-throughput, and they generate a huge amount of data. Thus, computer analysis has become indispensable to analyze data produced by them. Some commonly used computer aided techniques for high-throughput expression data analysis are as follows: pattern recognition, data extraction, data preprocessing, data integration, differential expression, clustering analysis, and gene expression time series analysis [38].

Various techniques are used for gene expression profiling. DNA microarrays are a high-throughput hybridization-based technique. A comparatively newer high-throughput technique, RNA-Seq, involves sequencing RNAs, and it has become widely adopted for studying gene expression profiling. Analyzing RNA-Seq datasets gives several advantages over DNA microarrays; for instance, the ability to detect SNPs (single-nucleotide polymorphisms), and alternative gene spliced transcripts [84].

The first commercially developed sequencing method was Sanger sequencing, developed by Frederick Sanger in 1977. The limitations of Sanger sequencing are mainly due to high sequencing costs and that they require a large amount of time per base to sequence. By way of contrast, modern sequencing platforms are relatively low cost, and are high throughout. Some examples are: Illumina NextSeq500 sequencing (US$42 per gigabase), SOLiD 5500 Wildfire (US$130 per gigabase), Pacific BioSciences RS II (US$1000 per gigabase) [28]. According to the National Human Genome Research Institute (NHGRI), the cost per genome is less than US$1000 (for a genome size of 3000 megabases) — excluding quality assessment, project management, and biological analyses expenses [2]. However, the impact of adding more samples on certain specific data analysis tasks is not well understood, nor are the trade-offs with time and costs.

Differential expression analysis has a pivotal role in gene expression profiling. In an organism, typically, all somatic cells contain the same set of genes. However, the functionality of the genes depends on how they are transcribed and translated in a cell. Genes being expressed differently between cell types was first observed using the DNA-RNA hybridization technique in 1968 [87]. As an example, say there are two patients, where one has tissue cells from an organ that are normal, and the other has a tumor. If there is a gene with a large difference in expression levels between the normal and diseased cells, this difference could be important to, e.g. diagnosis or treatment. Hence, analysis of the entire set of RNAs (the transcriptome) is fast becoming a key instrument for differential expression analysis. Generally, differential expression is used to help measure the regulation of genes under differing conditions, or in various groups of samples, or across multiple developmental stages, or for other research purposes [66]. RNA-Seq and DNA microarrays both have emerged as powerful platforms for differential expression analysis.

Another one of the most common data analysis techniques used with expression data is clustering. Clustering in general involves grouping together objects into clusters based on similarity or dissimilarity. That is, the primary goal of clustering is to divide objects into sets, called clusters, such that objects within clusters are highly similar to each other, and more diverse relationships exist between objects in separate clusters. Grouping objects (in our case, genes) also plays an important role in gene expression analysis. Typically, genes are defined to be similar if they have similar expression patterns. When genes are grouped together based on this type of similarity, this can provide evidence of related functionality, or that they are involved in some joint process. Cluster analysis can also be applied to find out different gene expression patterns on a small subset of genes [79].

Cluster analysis is a process of data exploration. Data is commonly presented in two formats: a data matrix and a distance matrix [91]. In the data matrix, genes are represented as rows, and samples are represented as columns, with entries being expression values. The distance matrix is used to show the similarity or the pairwise distances between two or multiple gene's expression values. Distances are created based on the data matrix, and a distance function (calculating the distance between two objects); for example, Euclidean distance. A higher distance between two genes expression patterns indicates lower similarity among them. Then, clustering can be performed from the distances in the distance matrix. Clusterings depend highly on the clustering algorithm used. Even the same datasets can give different results depending on which algorithms are being used.

The term clustering comparison refers to the process by which clusterings are analyzed and compared. A comparison can be used for two things: in particular, if a “correct answer” is known, then one could assess closeness between the true clustering and the predicted clustering. Additionally, comparisons might help to assess consistency of clusterings. Here consistency refers to how clusterings vary depending on the sets of samples used to construct them. Indeed, although cluster analysis is often used when no known ground truth information is known, with sufficient data, it is possible to assess the consistency of clusterings. This can be thought of as validating the clusterings in terms of approaching some local maxima. When clusterings are compared, some distance function is required (not to be confused with the distance function to create the clusterings). In this context, a small distance indicates similarity and large distance indicates dissimilarity between two clusterings.

The main objectives of this thesis are to:

1. Choose a cluster comparison distance to precisely compare K-means clusterings in order to validate

clusterings — primarily focused on RNA-Seq datasets. This proposed distance should be easy to apply and interpret in the context of gene expression clustering.

2. Investigate different algorithms for computing the cluster comparison distance exactly.

3. Test the effect of adding more biological samples to existing RNA-Seq expression datasets on K-means clusterings, in order to find a relationship between the number of biological samples and clusterings.

The third objective can be done with the help of the work on cluster comparison and the algorithms from the first and second objectives. This could help in understanding the trade-offs between the number of samples and the clusterings, which could help to reduce the number of samples needed for certain types of analysis. This analysis could provide an effective method to either reduce costs by requiring fewer samples in certain situations, or an understanding of the benefits additional samples would bring. Although the small number of datasets tested (three) limits the ability to make general conclusions, the results, plus the general approach provide an important contribution to research on gene expression profiling using RNA-Seq gene expression data by creating a methodology to assess how clusterings change as the number of samples increase.

This thesis is composed of five chapters. Chapter 2 contains the background, and begins by laying out the theoretical aspects of the research; it introduces the basic gene expression profiling techniques and provides a brief description of supervised (e.g. classification and regression analysis), and unsupervised (e.g. clustering) learning. Distance functions for clustering are also described in this chapter, which are one of the important parameters for a clustering algorithm. A study of existing clustering comparison methods is also presented. Chapter 3 describes the methodology, tools, and techniques that are used for this research. It describes the data normalization pipelines used for the RNA-Seq datasets analyzed. The three datasets used to test the methodology (mouse embryonic stem cell tissue, *Drosophila melanogaster* from multiple tissues and micro-climates, and mouse data from multiple tissues) are also described. This chapter also provides a detailed explanation of the different implementations of the comparison method, and the method used to assess consistency of clusterings with additional samples. Chapter 4 presents the results of the consistency of clusterings on all three datasets. A statistical analysis, an analysis of performance, and an assessment of the effectiveness of the clustering comparison results are presented in this chapter as well. The latter part of Chapter 4 discusses some implications of the results. Lastly, Chapter 5 gives a general review of this research work, describes the limitations of the proposed methods, and gives an overview of future directions.

2 Background

DNA is a macromolecule that encodes information in the form of chromosomes. Chromosomes copy information via DNA replication [35]. The central dogma of molecular biology states that the flow of information between molecules is mainly from DNA to RNA, and then RNA to protein (see Figure 2.1) [66, 35]. From a protein molecule, information will never pass back to nucleic acid. The central dogma can be divided into two parts. One is the transfer of information, and another is the conversion of information into other forms.

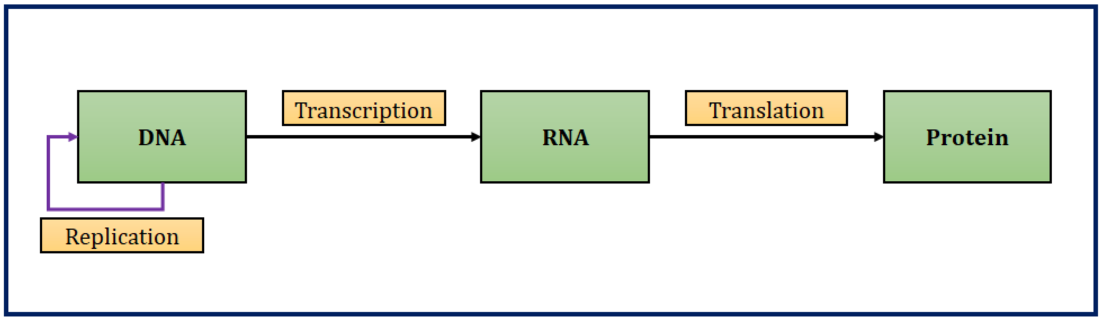


Figure 2.1: Central dogma of molecular biology.

Transcription is the process of creating RNA from DNA. Here, an enzyme called RNA polymerase reads the DNA, and produces an RNA molecule [35]. Then, in some genes of eukaryotic organisms, certain sections of RNAs, called introns, are removed, leaving the remaining sections, called exons. For protein coding RNAs, this produces messenger RNA (mRNA). In the process of translation, mRNA is converted into proteins, which is another type of macromolecule that consists of a chain of amino acids. The structure or the three-dimensional shape of a protein largely depends on the sequence of amino acids present in the polypeptide chain, and the structure largely dictates its function [94].

Both coding and noncoding RNAs (those that do not get translated to protein; e.g. tRNA, rRNA, etc.) together are called the transcriptome. Gene expression profiling is used to quantify the amount that each RNA sequence occurs in a set of cells from some tissue of an individual at a given time. This is usually multiple cells unless using single cell RNA-Seq. However, the amount of each RNA present varies from cell to cell, and tissue to tissue.

2.1 Gene expression profiling techniques

Different techniques are used for gene expression profiling. Early methods were low throughput and expensive [84]. More recently, high-throughput methods have become more common, which will be the focus of this work.

2.1.1 DNA microarrays

Variation in gene expression can be analyzed using microarrays. A DNA microarray or DNA-chip, is a glass slide that can contain thousands of microscopic DNA probes on its surface. DNA microarray technology uses the following two general steps [67, 84]: first, the production of a DNA microarray for a particular organism, and second, gene expression profiling of that organism's experimental cells to measure the transcriptome. DNA microarrays, often produced by chemical synthesis, can involve attaching short $20 - 30$ base pairs of single-stranded DNA, called probes, to the glass slides. Frequently, probes are constructed for each gene in an organism.

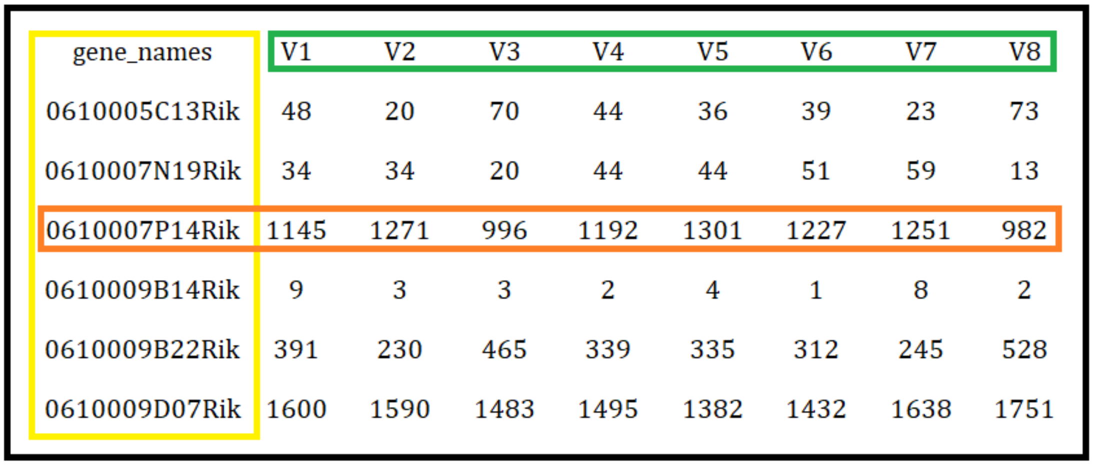
Two types of cells are commonly used for DNA microarray analyses. These are the control and targeted cells. Normal (or healthy) cells, and mutated (or diseased, or treated with a drug, etc.) cells of a particular organism are treated as control and target cells, respectively [67]. First, mRNA is extracted, then reverse transcribed into complementary DNA. Colour dyes (cyanide dyes) are applied to allow fluorescent intensity to be measurable. Then, these cells are placed onto the DNA microarrays (often separately). Hybridization occurs when one single-stranded copy of DNA binds to another complementary single-stranded DNA of one of the probes [26, 61]. After hybridization, a computer scanner is used to measure the amount of fluorescence label. Therefore, by looking at the intensity of the fluorescent labels, the quantity of each RNA sequence can be estimated. An alternative approach to using two types of cells are time trials, where samples are collected at multiple times of some biological process.

2.1.2 RNA-Seq

RNA-Seq, also called whole transcriptome shotgun sequencing (WTSS) [29, 84], involves sequencing RNAs in a sample using next-generation sequencing. This can determine which genes are active, and also estimate the amount of each mRNA produced at a certain time.

The first phase of RNA-Seq experiments is library preparation [84]. This involves RNA isolation, possible filtering, and then cDNA synthesis from the RNAs [28]. Sequencing the library is the second phase, which produces fragments of the cDNAs called reads. If an assembled genome already exists for the organism, the RNA-Seq reads can be aligned to the genome in a process called read mapping. Or if an assembled genome does not exist, the transcripts can be assembled into full RNAs in a *de novo* fashion. Then all transcript sequences are counted. Without normalization, it is not possible to compare expression levels between and within samples accurately. Li et al. [50] proposed a guideline for the selection of the most appropriate normalization methods for experiments. Various normalization methods can be used, such as the non-abundance method, the abundance method, or the inter-sample method [50].

After read mapping and normalization of all reads sequenced, the data matrix is generated (see Figure 2.2). Each row represents a single transcript, and each column represents an individual sample.



gene_names	V1	V2	V3	V4	V5	V6	V7	V8
0610005C13Rik	48	20	70	44	36	39	23	73
0610007N19Rik	34	34	20	44	44	51	59	13
0610007P14Rik	1145	1271	996	1192	1301	1227	1251	982
0610009B14Rik	9	3	3	2	4	1	8	2
0610009B22Rik	391	230	465	339	335	312	245	528
0610009D07Rik	1600	1590	1483	1495	1382	1432	1638	1751

Figure 2.2: The yellow coloured box indicates gene names, the green colour box represents the sample numbers, and orange colour box shows the count for that gene in each individual sample.

Data analysis is the final phase (for both RNA-Seq and DNA microarrays) of gene expression analysis. Firstly, it is common to assess differential expression, which involves calculating which genes are significantly different between the control and target sample sets. Clustering techniques are also widely applied in gene expression analysis. Cluster analysis can expose unknown connections among genes based on similar or correlated expression. Using gene expression profiling, clustering can also be helpful for pathway analyses of co-regulated genes [66]. Clustering will be discussed further in Section 2.2.2.

2.2 Machine learning

Machine learning and classification are important problems in engineering and scientific disciplines, and have been frequently applied to problems in biology, medicine, marketing, and many others. Classification involves the assigning of a discrete label to unlabelled data. Watanabe [86] defines a pattern “as opposite of a chaos; it is an entity, vaguely defined, that could be given a name.” A pattern could be a DNA fragment, an image of a handwritten cursive word, or it could be a speech signal. The recognition or classification can be categorized into two types: supervised classification and unsupervised classification. The aim of predictive or supervised learning is to map from input patterns to output patterns, given a (separate) set of a priori known input-output pairs called the training set. Input patterns consist of features, attributes, or covariates, in general. It could be of complex structure, a molecular shape, a graph, etc. [37]. The second main learning approach is descriptive or unsupervised classification, sometimes called knowledge discovery. Figure 2.3 shows a characterization overview of machine learning algorithms.

For both supervised learning and unsupervised learning approaches, parameters need to be set to develop a probabilistic model [60]. If a fixed number of parameters are used in the model, then it is called a parametric model. Supervised classification algorithms use parametric models. When the number of parameters increases depending on the sample size, then it is called a non-parametric model. Parametric models tend to be faster than non-parametric models [60]. However, for large datasets, strong prediction is often easier by using a non-parametric model, as it gives high flexibility to fit the data. However, it can often cause an overfitting problem, whereby a model is dominated by random samples or noise instead of by general patterns. Overfitted models are excessively complex; with such a model, a learned hypothesis may fit the training set very well, however it fails to generalize to new examples.

Cross-validation is an evaluation technique for validating a predicted model [60]. Learning approaches, like supervised learning, can use cross-validation for analyzing the outcome of a prediction. It can also be used to test how well a model can perform when real samples are applied. K-fold cross-validation divides the training sample into K equal sized parts. After that, it considers the first part for testing while using the remaining parts for training. Then, the second part is used for testing, etc., and the process continues for each part. Before applying testing samples, it is helpful to use cross-validation to check the performance of the model. It also helps to understand how results of the statistical analysis will generalize to the entire dataset.

2.2.1 Supervised learning

Two common supervised learning approaches are classification and regression, which will be briefly described.

Classification is a process to identify or categorize the features of a set of problems based on predefined labelled datasets. Classification varies depending on the features. If there are two different types of features, then it is called binary classification. If the features are classified into more than two types, then it is called multiclass classification. Multi-label classification can be mutually exclusive or not [60]. Features or attributes are often called explanatory variables. These features can be categorized in various ways; for instance, categorical (boy or girl), real-valued (temperature), integer-valued (frequency of arrival at a particular place), nominal (price range), or ordinal (rank of position).

A classifier is simply a mathematical function that is used to “solve” (by assigning labels to each item) a particular supervised classification problem. The goal of a classifier is to predict the outcome with maximum (or close to maximum) accuracy based on the labelled dataset. It can be formalized as calculating a function, $y = f(x)$, where y is the outcome, and x is in the predefined dataset. In machine learning, outcomes are often referred to by classes, and the predefined dataset types are called features or a feature vector [7].

Classifier performance evaluation is sometimes described as having “no-free-lunch”, meaning that no single method is suitable for all kinds of classification problems [89]. A classifier performance depends on the characteristics of the training set, and choosing the right classifier should be found according to the problem specified.

One common example of a classification problem is detecting whether an email is spam or not. In this case, the training set can be built with regular email (i.e. non-spam) and spam email. Here, large data samples help the classifier to distinguish an email into a correct class. Other classic classification applications are in the areas of computer vision, drug discovery and development, handwriting recognition, speech recognition, biological classification, etc.

A problem is called a regression problem when the features are nominal (scalar real-valued) variables. Statistical regression is the prediction of the relationship among the dependent variable with one or multiple independent variables [60]. Various techniques are applied to predict the impact of independent variables on the dependent variable. For example, the input features or the experimental setting or the environmental factors on a given problem can be set as the independent variables. A dependent variable is simply the outcome of the experimental result or the solution of a particular problem. Specifically, regression analysis helps to predict or assume the influence of a single independent variable to the dependent variable.

Regression analysis performance widely depends on the data generating process and the choice of method used. A regression model z can be defined as: $z \approx f(x, c)$, and approximation can be formalized as: $E(z \mid x) = f(x, c)$, where x is the independent variable(s), and the unknown factor is c. The function f must be defined between dependent and independent variables based on prior knowledge [60].

Linear regression: Linear regression creates a relationship between a dependent variable and independent variables. A regression line is used to build this relationship, which is a best fit straight line. An example is $y = a + bx + c_n$, where c_n is an error term for each n (number of data points); a and b are the intercept and slope, respectively. Simple linear regression occurs when the number of independent variables is one. Multiple independent variables are used for multiple linear regression [60].

Some other popular regression analyses are: polynomial regression and logistic regression. Polynomial regression is similar to linear regression except for the exponent on the independent variables. Higher order polynomial regression often has a lower error rate [60]. Depending on the independent variables or input features, logistic regression develops a model of probability for an event occurring. Logistic regression gives the estimation of a probability based on an event occurring or not occurring.

2.2.2 Unsupervised learning

Unsupervised learning is a predictive model, where datasets are not previously labelled, and also there is no defined output format. The outcome of this learning approach always depends on the observations. Unsupervised learning methodology looks for common patterns or structures in the testing data samples. It is a sort of a similar learning technique to how a human or an animal learns. It gains knowledge through experiences and from the environment. Unsupervised learning systems infer output without any prior knowledge or predefined labelled data.

Unsupervised learning methods develop a density estimation (constructs a probability distribution model based on unlabelled random responses) model for each input rather than defined outlines, like supervised learning [60]. Comparing the supervised learning technique to unsupervised learning, there are primarily two differences [32]. Because supervised learning predicts a single outcome for particular input variables, then it uses univariate probability density estimation. On the other hand, an unsupervised learning outcome is a vector of features; therefore, multivariate probability models are needed.

One of the most important types of unsupervised learning methods is clustering. The primary goal of clustering is to group data elements based on similarity or dissimilarity. It creates smaller subsets or partitions of related data. In various disciplines, clustering techniques are widely applied. For example, based on internet browsing history or interest, customers are being clustered by e-commerce sites to increase sales, and also to help to find target consumer groups. In biology, gene expression analysis uses clustering techniques to group sequences based on similar expression patterns. Clustering can be used as a stand-alone tool and also, can be used as a processing tool for other algorithms.

Clustering finds structure in a collection of unlabelled instances. For example, consider a collection of n objects, x_i, $1 \leq i \leq n$; each x_i is a p-dimensional feature vector. Then, one goal with clustering is to divide these n objects into k (a fixed number) of clusters such that the objects within a cluster are more "similar" to each other than objects between clusters. But what does this mean? There is no single answer — it depends on what distance is used to assess similarity on the data. For this reason, clustering is often referred to as an "art."

Two different types of clustering algorithms exist: strict partition clustering (each object is placed in some cluster and is not placed in more than one cluster) and overlapping clustering (cluster results may overlap) [93]. An example of strict partition clustering is K-means clustering, and an example of overlapping clustering is the expectation maximization (EM) algorithm. Some other popular clustering methods are hierarchical, self-organizing maps (SOMs), and mixture models.

Clustering is useful for identification of genes (in our case, these objects are genes) that are "working together" or that are co-related. This is beneficial for biological identification of distinct functional subgroups. A good clustering method will produce clusters with high intra-class similarity (objects that are similar belong to a single cluster) and low inter-class similarity (higher the distance for objects in different clusters) [93]. Seen as a graph where objects are connected if their distance is small, then after clustering, objects in the same cluster are connected densely, and only sparse connections exist between different clusters.

One unsupervised clustering method of interest is K-Means clustering. The goal of the K-means clustering problem is to partition a set of objects into K subsets or K clusters in order to minimize the within cluster variance [41]. The problem is NP-hard in general [6], and therefore any exact algorithms to solve it likely requires more than polynomial time complexity, and therefore heuristic algorithms are needed. A heuristic algorithm often called the K-means algorithm is commonly used for K-means clustering [52]. The approach of the K-means algorithm starts from an initial partition of the objects (e.g. genes) and proceeds by iteratively calculating the centers (means) of the clusters and then it reassigns each object to the closest cluster according to some measurement of distance such as Euclidean distance. This iteration continues until no more reassignments take place. From a computational perspective, this heuristic K-means algorithm is relatively efficient as it has a time complexity of $O(tkn)$, where n is the number of objects, k is the number of clusters, and t is the number of iterations, which is why the heuristic algorithm is more commonly used than the exact algorithm. Notice that this heuristic K-means algorithm is not deterministic in the sense that running it multiple times on the same input will often give different results.

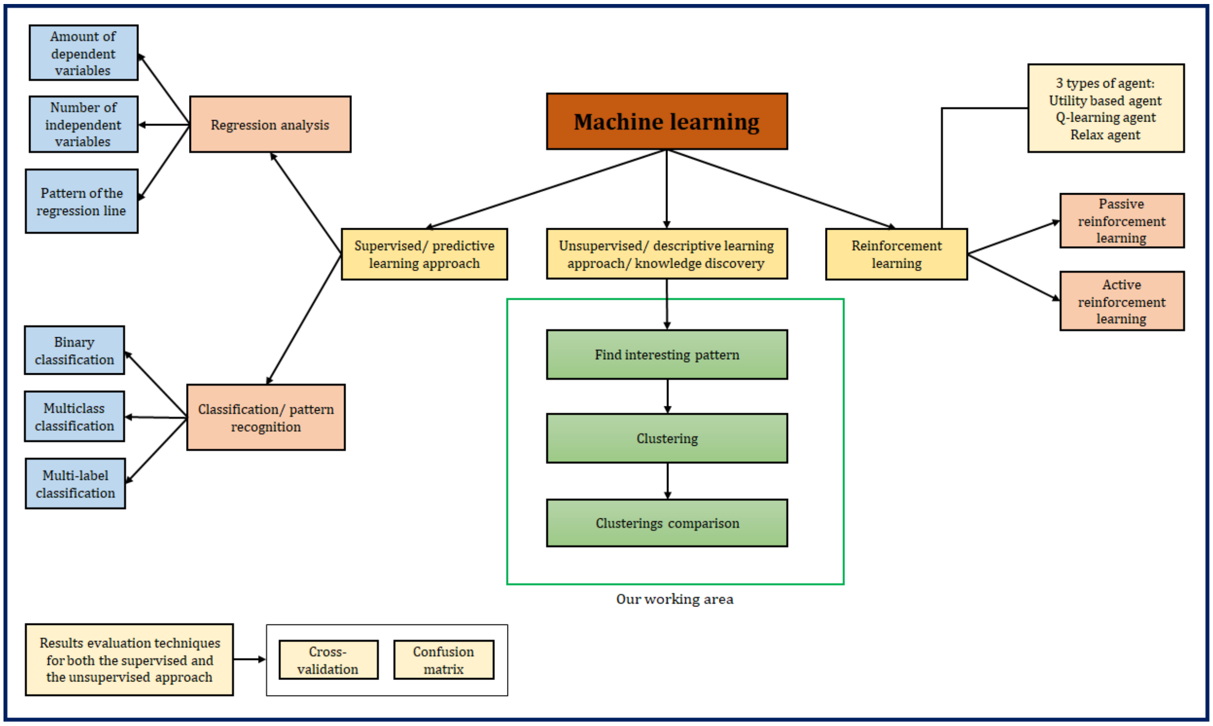

Amount of dependent variables
Number of independent variables
Pattern of the regression line
Regression analysis
Machine learning
3 types of agent:
Utility based agent
Q-learning agent
Relax agent
Supervised/ predictive learning approach
Unsupervised/ descriptive learning approach/ knowledge discovery
Reinforcement learning
Passive reinforcement learning
Active reinforcement learning
Find interesting pattern
Clustering
Clusterings comparison
Our working area
Binary classification
Multiclass classification
Multi-label classification
Classification/ pattern recognition
Results evaluation techniques for both the supervised and the unsupervised approach
Cross-validation
Confusion matrix

Figure 2.3: An overview of machine learning algorithms [60].

Distance functions

The objective of cluster analysis is to group objects based on similarity. Distance functions used for the purposes of creating clusterings will be called clustering distances (CD). These kinds of distance functions quantify the distance between two sets of objects, which can be used to measure similarity. Given a distance function, the goal with clustering is to place elements into clusters in order to minimize the intra-cluster distance, and maximize the inter-cluster distance (see Figure 2.4) [93]. Deza et al. [22] published a book named "Encyclopedia of Distances" which is one of the leading references of distance metrics, and covers most of the active research areas in distance functions.

For classification and regression problems, some commonly used distance measurement techniques are

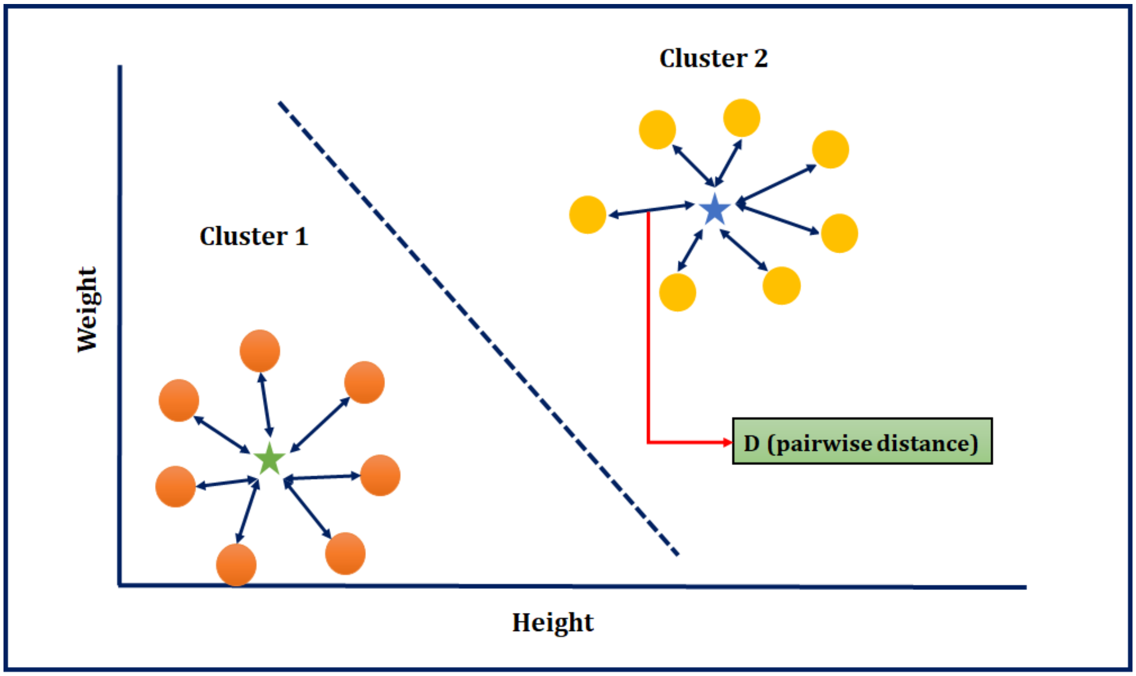


Figure 2.4: Each circle is a different person's height (x-axis) and weight (y-axis). There are two clusters represented by orange and gold colour balls respectively. Green and blue colour stars (cluster center) indicate the pairwise distances within each cluster.

Euclidean Distance, Minkowski Distance, Pearson's Distance, Hamming Distance, and Manhattan Distance, which are described next.

Euclidean distance

Euclidean distance is a simple distance measurement that can be used in Euclidean space. It measures the distance between points in n-dimensional space. In Cartesian coordinates, for two vectors $a = (a_1, a_2, a_3, \dots, a_n)$, $b = (b_1, b_2, b_3, \dots, b_n)$, the Euclidean distance between a to b (or b to a), is defined as follows:

$$D_E(a,b) = \sqrt{\left((a_1 - b_1)^2 + (a_2 - b_2)^2 + \cdots + (a_n - b_n)^2\right)}.$$

For example, say a and b are both vectors with five components, $a = (5, 6, 9, 10, 18)$ and $b = (3, 8, 9, 8, 20)$. Then the distance $D_E(a,b) = \sqrt{\left((5-3)^2 + (6-8)^2 + (9-9)^2 + (10-8)^2 + (18-20)^2\right)}$. Therefore, $D_E(a,b) = 4$.

Another variant of Euclidean distance is Euclidean squared distance,

$$D_E^2(a,b) = \left((a_1 - b_1)^2 + (a_2 - b_2)^2 + \cdots + (a_n - b_n)^2\right).$$

Euclidean squared distance and Euclidean distance both use almost the same base equation. Although, Euclidean distance is suitable for small distance calculations, using the Euclidean squared distance in clustering algorithms is faster in comparison to using Euclidean distance [60].

The Euclidean distance from the origin to a vector is a called the Euclidean norm or Euclidean magnitude [16]. That is, the Euclidean norm of $a = (a_1, a_2, \ldots, a_n)$ is: $||a|| = \sqrt{a_1^2 + a_2^2 + \cdots + a_n^2}$.

Manhattan (city-block) distance

The Manhattan distance or city block distance is defined as the sum of the absolute differences of each component of the two points in Cartesian coordinates [14]. Here, the distance is equal to the length of all shortest paths connecting to a and b along horizontal and vertical segments. For n-dimensions data points, $a = (a_1, a_2, a_3, \ldots, a_n)$ and $b = (b_1, b_2, b_3, \ldots, b_n)$, then the Manhattan distance is:

$$D_{Mn}(a, b) = (|a_1 - b_1| + |a_2 - b_2| + \cdots + |a_n - b_n|) .$$

For example, given two points $a = (2, 2)$ and $b = (3, 1)$, the Manhattan distance between a and b is $D_{Mn}(a, b) = (|2 - 3| + |2 - 1|) = 2$

The name Manhattan distance came from the grid layout of most streets in Manhattan Island [14]. Often Manhattan distance is called the L_1 distance (norm), which is the summation of the absolute values of two sides in a right angled triangle. Manhattan distance is employed for discrete frequency distribution. For example, to compare the positional distribution of hexamers in RNA splicing, Manhattan distance is used [51]. It is also used in sparse sampling (also known as compressed sensing) which is a signal processing technique in an undetermined linear system for acquiring and reconstructing a signal.

Minkowski distance

Minkowski distance is a generalized form of both Euclidean distance and Manhattan distance [14]. It is a distance function which can be defined as the norm (length of the vector) in a norm vector space. The general form of Minkowski distance is called L_m distance. The Minkowski distance of order m between two n-tuples points is given below: for $a = (a_1, a_2, a_3, \ldots, a_n)$ and $b = (b_1, b_2, b_3, \ldots, b_n)$,

$$D_{Mi}(a, b) = \left(|a_1 - b_1|^m + |a_2 - b_2|^m + \cdots + |a_n - b_n|^m\right)^{1/m} .$$

Another variation of Minkowski distance is weighted Minkowski distance,

$$D_{Mi}(a, b, w) = \left(w_1|a_1 - b_1|^m + w_2|a_2 - b_2|^m + \cdots + w_n|a_n - b_n|^m\right)^{1/m} ,$$

where weights $w = (w_1, w_2, w_3, \ldots, w_n)$ are chosen based on the application.

Pearson's distance

Based on Pearson's correlation coefficient, Pearson's distance is calculated between two n-tuples to measure their linear relationship [22]. For $a = (a_1, a_2, a_3, \ldots, a_n)$ and $b = (b_1, b_2, b_3, \ldots, b_n)$, then the Pearson's distance is:

$$D_{Pearson} = 1 - r(a, b),$$

where $r(a, b) = \frac{\sum_{i=1}^{n} (a_i - \overline{a})(b_i - \overline{b})}{\sqrt{\sum_{i=1}^{n} (a_i - \overline{a})^2} \sqrt{\sum_{i=1}^{n} (b_i - \overline{b})^2}}$, with $\overline{a}$ the mean of values in a, and similarly with b, is Pearson's correlation coefficient between a and b.

Canberra distance

Canberra distance is a weighted form of Manhattan distance developed and improved by Williams Lance and Adkins in 1966 [44]. It measures distances between scatter data or to group individuals from an origin. For n-dimensions data points, $a = (a_1, a_2, a_3, \ldots, a_n)$ and $b = (b_1, b_2, b_3, \ldots, b_n)$, then the Canberra distance is:

$$D_{Ca}(a, b) = \frac{|a_1 - b_1| + |a_2 - b_2| + \cdots + |a_n - b_n|}{|a_1 + b_1| + |a_2 + b_2| + \cdots + |a_n + b_n|}.$$

χ^2 distance

To calculate the distance between two histograms, a distance metric called χ^2 distance can be used [39]. In the χ^2 distance, two histograms (discrete probability distributions) should have an equal number of bins to calculate the difference between them. It is often used for document classification in computer vision (which is called the bag-of-words model). A χ^2 distance is a weighted form of Euclidean distance. Given an observed value $a = (a_1, a_2, a_3, \ldots, a_n)$, and an expected value $b = (b_1, b_2, b_3, \ldots, b_n)$ having n bins, the χ^2 distance between a and b is as follows:

$$D_{\chi^2}(a, b) = \frac{1}{2} \sum \left((a_i - b_i)^2 / (a_i + b_i) \right).$$

Hamming distance

The Hamming distance function is used to calculate the distance between two categorical variables [31]. Here, 0 indicates a similar feature of two categorical variables, and 1 otherwise. The Hamming distance is obtained after adding all those differences. For example, if a and b are two categorical variables of d features, then the Hamming distance between a and b is as follows:

$$D_H(a, b) = \sum_{i=1}^{d} (a_i \Leftrightarrow b_i),$$

where $(a_i \Leftrightarrow b_i) = 0$ if and only if a_i and b_i indicate similar features, and 1 otherwise.

Kullback–Leibler divergence

In machine learning, Kullback–Leibler (KL) divergence is used to measure the distance between two distributions [43]. It measures the divergence of a probability distribution from a reference probability distribution. The KL divergence score varies from 0 to 1. Here, 0 indicates that the two distributions are the same or there is no difference between them, and 1 depicts that comparing distribution shows a different pattern than the reference distribution. If P and Q are two probability distributions, then the KL divergence between P and

Q is:

$$D_{KL}(P||Q) = \int_{x \in D} p\left(x\right) \log \frac{p\left(x\right)}{q\left(x\right)} dx,$$

where, $p(x)$ and $q(x)$ are the probability density of P and Q respectively, and x is in the range of domain D.

2.3 Cluster comparison

A cluster comparison is a measure of similarity between two different clusterings. The two clusterings could have been produced by the same algorithm using different parameters, by using different algorithms altogether, or using the same algorithm by using different subsets of the data. It could ultimately be used as a technique to validate clusterings.

It is necessary to clarify exactly what is meant by clustering comparison. As previously mentioned, a group of similar instances that have been grouped together is called a cluster. When multiple clusters are created from a dataset, this is a clustering. The comparison between two clusterings is called a clustering comparison. Just as distances are used to create a clustering, distances can also be used to compare two clusterings. The distance functions used to compare two clusterings are called cluster comparison distances (CCD). It is important to keep in mind the differences between CDs (clustering distances) and CCDs. In this context, a lower distance score indicates that two clusterings are similar. This score can be used as an indication of a clustering algorithm's quality if a correct answer is known.

Meilă [56] surveyed different cluster comparison methods. Here, only cluster comparisons of strict partition clusterings will be discussed. Let X be a set of n elements: $X = \{x_1, x_2, x_3, \ldots, x_n\}$. Next, let G_1 and G_2 be two clusterings, where $G_1 = \{P_1, P_2, P_3, \ldots, P_k\}$ and $G_2 = \{Q_1, Q_2, Q_3, \ldots, Q_l\}$. Each element of G_1 and G_2 is a subset of X, and the elements of G_1 or G_2 are disjoint sets, such that $\bigcup_{i=1}^{k} P_i = X$, and $\bigcup_{i=1}^{l} Q_i = X$. Thus, G_1 and G_2 are sets of disjoint sets of elements of X whose union is the set of all elements. For example, if $X = \{x_1, x_2, x_3, x_4\}$, and $k = l = 2$, then one possible clustering could be: $G_1 = \{P_1, P_2\}$, where $P_1 = \{x_1, x_2, x_3\}$ and $P_2 = \{x_4\}$, and another possible clustering could be $G_2 = \{Q_1, Q_2\}$, where $Q_1 = \{x_1, x_3\}$ and $Q_2 = \{x_2, x_4\}$. A clustering comparison distance measures the difference between the elements in G_1 and G_2.

A similar task is cluster evaluation which can be used to assess the quality of clusterings. Evaluating or validating clusterings is a difficult task. As clustering is often used as an unsupervised method (if the ground truth is unknown), it is hard to evaluate a clustering [60]. Popular approaches of evaluating clusterings fall into two groups: internal evaluation and external evaluation. An internal evaluation technique is only evaluated within a clustering (e.g., Dann index). External evaluation can be applied when a predefined or reference clustering is available (e.g., purity).

There are several techniques used for clustering evaluation and cluster comparison, which are discussed in the next section.

Cluster comparison distances

One natural approach for comparing clustering is pair counting. This classifies each pair of objects in discrete categories and then counts the results. A pair of unordered objects can be classified in four different ways: 1. the pair appears in the same cluster of both clusterings, 2. the pair does not appear in the same cluster within both clusterings, 3. the pair appears in the same cluster of the first clustering, and in different clusters of the second clustering, or, 4. the pair appears in the same cluster of the second clustering, and different clusters of the first clustering. Several mathematical measures are found in the literature for cluster comparison based on the pair counting technique, such as Rand index [68], and χ^2 coefficient [64].

The Rand index is the ratio of the number of elements where both clusterings put them in the same cluster, plus the number that are placed in different clusters of both clustering, divided by the total number of pairs. The range of Rand index varies from 0 (all pairs that appear in the same cluster of one clustering appear in different clusters of the other clustering, and vice-versa) to 1 (each pair either appears in the same cluster of both clusterings, or in different clusters of both clusterings). If it is preferred, one minus the Rand index gives smaller values for similar clusterings. Different variations of Rand index exist in statistics; for example, adjusted Rand index is commonly used [75]. Albatineh et al. [5] give a list of 28 comparison measures based on Rand index and pair counting. However, some of the measures become equivalent after making some changes to the pair counting technique [85].

Meilă [56] presents a Classification Error (CE) metric between two clusterings. Let G_1 and G_2 be two clusterings where G_1 has K clusters, G_2 has K' clusters and $K \leq K'$, and n be the number of elements clustered. Then $D_{CE}(G_1, G_2) = 1 - \frac{1}{n} \max_\sigma \sum_{k=1}^{K} n_{k,\sigma(k)}$, where σ is any injective mapping function to match clusters of G_1 to G_2 (injective means no two clusters of G_1 map to the same cluster of G_2). The author mentioned that despite the at least $K!$ possible injective mappings, the maximal bipartite matching algorithm in graph theory can calculate the distance exactly in polynomial time. Meilă also points out that CE distance is simple and intuitive, especially when clusterings are similar.

Probabilistic approaches use likelihood to compare clusterings. One commonly applied technique is to compute the distance between two probability distributions. One method for this is called EMD (earth mover's distance) [48, 72]. Here each clustering represents a distribution. To compare the distributions, it first fragments each of them. The EMD can then be calculated by measuring the distances between each fragment of the two distributions. To get the final EMD between two distributions, it adds all the distances between each fragment.

Mutual information of two clusterings measures the number of common objects obtained from one clustering compared to the other. The mutual information is inspired by the idea of entropy theory (calculating the missing information) [20]. One of the variations of mutual information is adjusted mutual information (AMI) [81]. The adjusted mutual information measure is often used to calculate the similarity between two clusterings. It measures two clusterings based on their distribution of results. If two distributions are randomly distributed, then AMI returns 0; otherwise, it returns 1 which means the two distributions are identical.

Another common technique, named word mover's distance (primarily focused on text matching), measures the relatedness of two words by measuring the closeness in meaning [83]. One of the pioneer researchers of distribution-based clustering comparison is Marina Meilă [56, 33, 55, 57]. Meilă introduced more partitioning properties in cluster comparison using entropy theory. D. Zhou et al. [93] proposed a clustering comparison metric for global optimization, inspired by the Mallows distance — computing the distance between two distributions. For clustering comparison, the authors use both strict partition clustering and overlapping clusterings.

Clustering evaluation metrics

Validity measure, or v-measure, is an external measure of clustering, and it uses a ground truth clustering. For a given clustering, completeness and homogeneity metrics are calculated. A clustering satisfies the completeness if all class members (data points) are in the same cluster compared to the reference class. A clustering satisfies homogeneity when each of the class members are in the same class label and in the same cluster compared to the reference clustering. Rosenberg et al. [70] combined these into v-measure to evaluate a clustering. Its scores are calculated by using the harmonic mean of homogeneity and competence values.

Purity is another external measure of clustering quality for overlapping clusterings [60]. It measures the intra-cluster similarity. Purity only considers majority clusters and numbers of objects of those majority clusters. Then, it counts the maximum number of objects in each cluster, and adds those maximum counts. Therefore: $purity = \frac{1}{N}\sum_{i \in k} max_j\,(n_{ij})$, where, k is the set of clusters, N is the total numbers of objects, and n_{ij} indicates the numbers of objects j in a cluster i. Its scores vary from 0 (poor similarity) to 1 (good similarity). However, purity never penalizes different cluster sizes in clusterings, and a major drawback of purity is its often high value.

Set matching or set overlaps are primarily used for document classification. Set matching defines a set of objects which are common to both clusters. A classic example of a scoring metric for set matching is F-score (defined below). There are other measurement tools that are available based on set matching, for example, Van Dongen-Measure [23, 82]. For error counting or to test the accuracy, F-score is commonly used in set matching [47]. F-score is used as a binary classification — positive or negative — and is also called the harmonic mean of precision and recall. Here, precision is the ratio of the numbers of correctly identified results divided by all positive results returned by a classifier. The recall is the ratio of the numbers of correctly identified results divided by the number of true positives plus false negatives. Then, $F\text{–}score = \frac{2(precision \times recall)}{precision + recall}$. The F-score varies from 0 (total incorrect prediction) to 1 (correct prediction). Based on the principle of precision and recall, another clustering comparison metric — Fowlkes-Mallows score — is generally used. The Fowlkes-Mallows score depends on the predefined clusterings, which is used as a reference or a benchmark [27]. Fowlkes-Mallows scores can be defined as a geometric mean of precision and recall: $FMS = \sqrt{precision \times recall}$.

The R^2 coefficients is defined by the summation of within cluster and between cluster sum of squares, which can be used as an evaluation metric for unsupervised clusterings [74].

Silhouette coefficient is another clustering evaluation metric [71]. It is a graphical tool which is used to test the validity and consistency of clusterings. Such a comparison depends on the variance of clustering elements. A lower variance between cluster elements and higher variance within clusters is expected for better similarity. Predefined labelled or reference output is not necessary for silhouette index score. However, the computational cost is high for higher cluster size. Its scores vary from -1 to $+1$, where -1 indicates incorrect clustering, 0 and $+1$ represents overlapping and dense clustering, respectively.

David et al. [21] introduced an internal clustering comparison metric named the Davies–Bouldin index (DBI). The DBI index primarily focuses on the assignment of objects within a cluster. It measures the distance of each object in a cluster from the centroid of the cluster. By comparing the distance within a cluster, the DBI validates clusterings — how well clusters are created. The norm distance function is generally used to calculate the distance from a cluster centroid to each data object.

Dunn [25] proposed another clustering evaluation metric which is referred to as the Dunn index (DI). The Dunn index calculates the mean difference of within-cluster objects compared to between cluster objects. A high score of DI indicates that objects within a cluster are densely connected, and sparse connections exist between different clusters. One of the limitations of DI is high execution time as the cluster size and numbers of data points increase.

Based on the dispersion of cluster objects, the Calinski Harabasz (CH) measure was developed for comparing clusterings [17]. The CH metric is calculated by the ratio of within and between cluster dispersion scores (variance of a distribution). A higher CH index shows that objects within a cluster are densely connected and are well separated by the other cluster objects.

Other frequently applied clustering evaluation metrics are the following unsupervised evaluation techniques: Gamma and Tau [13], C-Index [34], Gap statistics [76], I-Index [54], S Dbw (separation and density based) [30], DBVC (density based cluster validation) [59], and the following supervised evaluation techniques: B-Cubed evaluation [12], Set matching purity [92], Gini-based evaluation [73].

Other related works

To characterize a clustering, and to overcome the limitation of previously existing clustering comparison techniques, E. Bae et al. [11] proposed a density profile to measure the similarity. The proposed comparison method is ADCO (Attribute Distribution Clustering Orthogonality), which is inspired by the area of data mining. Here, the similarity between two clusterings depends on the prediction model, and two clusterings are more likely similar if the two predictive models are similar. Additionally, other methods are proposed, for example, to identify structural dissimilarity, and to compare non-overlapping clustering methods.

To minimize the error rate of different clusterings, Backer and Jain [10] suggest various partitioning techniques for different clustering algorithms. For clustering validation, S. Monti et al. [58] used unsupervised learning algorithms (K-means, SOM, etc.) on DNA microarrays of gene expression datasets. They show that the experimental results (both simulated and real data) are biologically meaningful for cluster analysis.

2.4 Tools for machine learning and for clusterings comparison

Due to the diversity and complex nature of bioinformatics problems, it is important to choose the right tool for the given task [24]. In our case, the tool often dictates a programming language. The most popular modern interpreted scripting languages are Perl, Python, and R. There are many reasons for using these languages: for example, memory management, code readability, dynamic type system, and the ability to build prototype programs in an interpreted and extensible environment [24].

For scientific computing, a number of open-source frameworks are commonly used. For example, NumPy, SciPy for Python, Ruby on Rails, etc. Besides that, many other structured frameworks particularly designed for bioinformatics are available, such as BioPython (Python), BioJava (Java), BioConductor (R), BioPERL (Perl), and BioRuby (Ruby). These structured frameworks are well documented, rigorously scrutinized, and involve an enthusiastic community providing regular improvement.

A scripting language is also often used for automating the execution of tasks using the run-time environment. They can combine complex programs or API calls, and can be used as a domain specific language.

The main difference between a compiled language and a scripting language is its compilation step. To run a program, compiled languages require a compiler to convert into some other format of code, either machine code or some higher-level intermediate code such as Java's bytecode. On the other hand, scripting languages can execute a program without compiling the entire program in advance. Scripting languages can make it easier for the developer to modify functionality. In the case of Python, it can work both as a compiled language (use CPython for implementation, convert source code into bytecode, then return the bytecode in a virtual machine), and a scripting language as well.

Here two main platforms are used: VLFeat (MATLAB) and Python scikit-learn. R is also used for some statistical analysis.

VLFeat

VLFeat is an open source MATLAB library [80]. It is a collection of cross-platform classes and packages, primarily developed for popular computer vision algorithms, with a special focus on local feature extraction, and the matching of images on large datasets. Of interest, VLFeat has implemented a large pool of machine learning algorithms. For example, the following statistical methods are included: GMM (Gaussian mixture model) using the expectation maximization algorithm, K-means, SVM (Support vector machine), KD-trees, etc. There are many visual features implemented, for instance, covariant detectors, HOG (histogram of oriented gradients), SIFT (scale invariant feature transform), dense SIFT, Fisher vector, and many others. VLFeat provides a C wrapper function for other programming languages and supported on multiple platforms, for example, Windows, macOS, and Linux. The MATLAB interface is an easy way to use the VLFeat library that allows to run the same algorithm on different platforms.

Python scikit-learn

Scikit-learn [65] is a toolkit of SciPy (Scientific Python). It is primarily developed for machine learning algorithms. Data distribution, filtering, aggregation, and classic machine learning algorithms are implemented in the scikit-learn library. For instance, classification clustering algorithms — K-means, K-nearest neighbor (KNN), hierarchical clustering; regression algorithms, support vector machine (SVM), etc. Standardized tools are available for data preprocessing, and it is easy to create one's own model to fit the test data using scikit-learn. Multiple library functions are available for evaluating model's performance; for example, classification matrices (accuracy score, classification report, confusion matrix), regression matrices (mean absolute error, mean squared error, R2 score), clustering matrices (adjusted rand index, homogeneity, V-measure), and cross-validation.

3 Methodology

In this chapter, first, the data and the procedure used to preprocess the datasets will be described. Then, the methodology and metric used for the clustering comparison will be given, along with the algorithms for its calculation. Then, the methodology for evaluation of the RNA-Seq datasets will be described.

3.1 Datasets

Three major datasets were used in this study: a mouse embryonic stem cell tissue dataset, a mouse multi-tissue dataset, and a *Drosophila melanogaster* dataset from multiple tissues and microclimates. The following sections describe the data preprocessing pipeline and the standard tools used.

Mouse embryonic stem cell tissue

The first dataset is RNA-Seq gene expression data from mouse embryonic stem cell tissue containing 78 samples [62]. The data is available from the European Bioinformatics Institute (EMBL-EBI) (EBI's array express accession codes: E-MTAB-3234 and E-MTAB-2830). The $mm10$ version of the mouse genome annotation files from Ensembl was used as a reference. Illumina sequencing was used.

Mouse multi-tissue

Preprocessed mouse multi-tissue data was used (GEO accession: GSE108990) [36]. It contains 70 samples with 23 lung tissue samples, 23 liver tissue samples, and 24 kidney tissue samples with 24485 genes in each sample. Illumina sequencing was used. The already normalized read counts are available in GEO.

Drosophila melanogaster

RNA-Seq data was used from *Drosophila melanogaster* consisting of 64 samples from two divergent microclimates: heads of 32 isofemale fly lines, and whole bodies of 32 isofemale fly lines [90], with 16 samples from each microclimate (GEO accession: GSE104073). For mapping, a *Drosophila* reference genome [4] from Ensembl was downloaded. Illumina sequencing was used.

3.2 Preprocessing

Since we are looking for clustering differences only within each of the three dataset, and they are each independent, different preprocessing methods are used for each of the three different datasets. A systematic study of the effect of different preprocessing methodologies on clusterings is beyond the scope of this work. The two mouse datasets (embryonic stem cell and multi-tissue) were already preprocessed, and we preprocessed the *Drosophila melanogaster* dataset. Major steps involved in the preprocessing of all three datasets are described below.

Preprocessing mouse embryonic stem cell data

This dataset was preprocessed by Katie Ovens in the McQuillan Lab [62]. The preprocessing she used is described next.

It was necessary to check the quality of raw sequences, which was done using FastQC [1]. FastQC does some initial quality control validation of raw sequences as it helps to identify probable problems or biases before drawing any biological conclusions. A Java application, Trimmomatic [15], was applied after FastQC to filter out low quality reads.

TopHat (version 2.0.13) was used to align RNA-Seq reads to a reference genome [77]. For the alignments, TopHat uses an alignment program, Bowtie (version 2.1.0) [46], which is a high-throughput short read alignment tool for large genomes. Both tools are popular partially due to their high computing efficiency, use of parallel processing, and low memory usage. The SAM (sequence alignment map) format is used for output, which is a generic format for storing read alignments against a reference genome for short and long reads.

A Python library, HTSeq, was then used for analyzing the high-throughput sequencing data. It can be used to count the number of discrete transcripts for each sample [8].

Normalization is needed to remove biases, while ensuring minimal impact of bias on the result. Without normalization, it is not possible to compare expression levels between and within samples accurately. Trimmed mean of M values (TMM) normalization was used to correct for library size [69]. The edgeR (a batch normalization technique) package, depend on the total read counts among all samples, from Bioconductor was used to normalize with TMM.

Preprocessing mouse multi-tissue data

A quality control score of 20 was used to filter out low quality reads. Adapters were removed with cutadapt [53]. STAR (Spliced Transcripts Alignment to a Reference) is an alignment tool that was used for mapping and alignment of this dataset. It has a high computation speed; however, STAR needs more memory than TopHat [40]. Reads per kilobase million (RPKM) was used for the postalignment quantification.

Preprocessing *Drosophila melanogaster* data

The downloaded reference genome contains the reference index for Bowtie and TopHat. The index for Bowtie was built using the "$bowtie2-build$" command, which uses the genome FASTA file. To prepare the reference index, TopHat uses genomic annotation file format. The annotation file was downloaded from Ensembl (`https://uswest.ensembl.org/info/data/ftp/index.html`) in GTF (gene transfer format) [40].

The same pipeline to analyze the RNA-Seq data from Trapnell et al. [78] was used. Since the samples are paired-end, in TopHat, the strand-specific RNA-Seq protocol for sequence alignment was followed. After running TopHat, Cufflinks (version 2.1.1) was used here to quantify transcript counts. The Cufflinks suite uses the FPKM (Fragments Per Kilobase Million) normalization method to express transcript levels. The output folder contained the FPKM-tracking files of gene-level, transcripts, and isoforms with confidence. Then the FPKM_tracking files from the Cufflinks output were merged to obtain the expression levels of all samples (see e.g. Table 3.1). In Figure 3.3, the mean sorted log distribution of *Drosophila melanogaster* gene expression data is summarized.

A Linux server (OS) was used for preprocessing the *Drosophila melanogaster* dataset. Sample code is provided in the appendix section.

Table 3.1: A small subset of the gene expression data from the *Drosophila melanogaster* dataset.

gene_id	**1S_WB**	**2S_WB**	**3S_WB**	**4S_WB**	**5S_WB**	**6S_WB**	**7S_WB**
CUFF.1	13.5844	6.11953	8.19355	9.70748	19.1139	12.1656	7.32032
CUFF.2	2.88508	3.1006	2.85698	4.22418	3.62807	2.56929	4.35421
CUFF.4	10.4157	4.21681	11.0639	4.02679	2.57215	5.55237	10.4282
CUFF.3	14.34	20.2349	20.627	18.4203	1.57383	25.5347	12.0123
CUFF.5	7.85974	4.66579	17.5395	26.9996	7.73811	9.5218	16.8399
CUFF.7	7.60263	9.64131	11.6952	3.88352	2.88756	25.8311	2.99667
CUFF.6	5.68415	34.3546	74.0531	23.5303	10.4314	64.6895	17.2517
CUFF.9	20.9965	3.26655	4.93884	3.99626	20.3058	2.93173	4.13324
CUFF.10	18.1999	84.5887	26.9367	21.3088	18.8945	31.079	2.57711

3.3 Cluster comparison metric

Cluster comparison methods can compare two clusterings generated by a given algorithm. Such a method of comparison can answer the question of how much the clusterings can change when using different subsets of the samples to construct them. In this section, a cluster comparison metric is described, as well as three algorithms which calculate it. The metric is used for comparing two sets of clusterings, both with k clusters. It is a variant of Classification Error (CE) defined in Section 2.3, and is similar to Earth Mover's Distance. It is chosen to be appropriate for the testing of consistency of RNA-Seq datasets.

The CCD (cluster comparison distance) method is as follows: given two strict partition clusterings G_1, G_2 with $|G_1| = |G_2| = k$ (equal number of clusters), then what is the minimum number of elements that can be moved from one cluster to another in G_1 to produce G_2? This method is meant to quantify precisely the differences between the two clusterings. It could just as easily be defined where G_1 and G_2 contain different numbers of clusters, as with CE. But the RNA-Seq tests performed here only require the same number of clusters in both.

Formally, let $X = \{g_1, g_2, g_3, \ldots, g_n\}$ be the universe (the elements to be clustered, such as genes). Let exp be a function that assigns every element in the universe (gene) to an m-tuple, of its expression values in m different samples. Let $G_1 = \{P_1, P_2, P_3, \ldots, P_k\}$, $G_2 = \{Q_1, Q_2, Q_3, \ldots, Q_k\}$ be two clusterings of X [56]. That is, both G_1 and G_2 are partitions of X. So each element of $\{g_1, g_2, g_3, \ldots, g_n\}$ appears in exactly one set of $\{P_1, P_2, P_3, \ldots, P_k\}$ and exactly one set of $\{Q_1, Q_2, Q_3, \ldots, Q_k\}$, and $\bigcup_{i=1}^{k} P_i = X$ and $\bigcup_{i=1}^{k} Q_i = X$. Given two subsets P, Q of X, define the distance between P and Q to be: $D(P, Q) = |P - Q|$ (where the $-$ is set difference) i.e. how many elements are only in P and not in Q.

Therefore, the distance between every pair of clusters P_i and Q_j of the two clusterings can be calculated: $|P_i - Q_j|$, for $i = 1, \ldots, k$ and $j = 1, \ldots, k$; where k is the number of clusters.

Although the distance function D can be used to compare a single cluster from G_1 to a single cluster from G_2, when comparing G_1 to G_2, it is not clear which clusters from G_1 should map onto a cluster from G_2. Consider a function called θ that is a permutation from G_1 to G_2 (mapping each cluster of G_1 to a unique cluster of G_2). First, the distance between G_1 and G_2 with the mapping θ is defined to be $D(G_1, G_2, \theta) = \sum_{i=1}^{k} D(P_i, \theta(P_i))$. Therefore, given a specific mapping between clusters θ, $D(G_1, G_2, \theta)$ adds up the distances between the mapped clusters. Then, the distance between G_1 to G_2:

$$D(G_1, G_2) = \min_{perm\ \theta} (D(G_1, G_2, \theta)). \tag{3.1}$$

It is possible to compare two arbitrary clusterings G_1 and G_2 using Equation 3.1. Indeed, at most this many elements of G_1 need to moved to a different cluster to produce G_2. Furthermore, if there was a smaller number of elements that could be moved, then θ would not be minimal. Hence, $D(G_1, G_2)$ describes precisely the smallest number of elements of G_1 that need to be moved to produce G_2. Moreover, a minimal θ in Equation 3.1 describes an optimal correspondence, or labelling, of the clusters.

Additionally, this method, satisfies the properties to be a metric. Those properties are [56]: given three arbitrary clusterings G_1, G_2, and G_3 (a) non-negativity — $D(G_1, G_2) \geq 0$, (b) $D(G_1, G_2) = 0$ if and only if $G_1 = G_2$, (c) symmetry — $D(G_1, G_2) = D(G_2, G_1)$, and (d) triangular inequality — $D(G_1, G_3) \leq D(G_1, G_2) + D(G_2, G_3)$, as changing G_1 to produce G_3 can be obtained by changing G_1 to G_2 and then G_2 to G_3.

For clustering, an appropriate distance should be chosen based on context. Similarly, according to Meilă [56], “Just as one cannot define a “best” clustering method out of context, one cannot define a criterion for comparing clusterings that fits every problem exactly”. Hence, an appropriate distance should be chosen for the application of RNA-Seq consistency analysis. The chosen metric in Equation 3.1 is quite similar to Classification Error (CE). The injective function in CE is similar to θ in Equation 3.1. Any injective function used for CE where G_1 and G_2 both have k clusters is a permutation. In Equation 3.1, differences between pairs of clusters are added to know exactly the differences in the clusterings. In contrast, in CE, common elements are counted, and maximized, which is essentially the same. Because of this difference, the definition of CE scales values to between 0 and 1, and then subtracts from 1; so smaller distances are given for more similar clusterings. The metric is also similar to Earth Mover’s Distance which is “proportional to the minimum amount of work required to change one distribution into the other” [19]. Also, the minimum number of genes needed to be moved from G_1 to produce G_2 is simple to interpret. For RNA-Seq consistency analysis, this distance will make it possible to quantify how many genes are different between a clustering made with X samples vs. one made with Y samples. This will also help in understanding the incremental benefit of adding more samples. In other words — is changing Z genes in the clustering worth collecting more samples?

3.4 Methodology and implementation of cluster comparison on RNA-Seq data

Here, the methodology used for cluster comparison (using Equation 3.1) on the three RNA-Seq datasets is described.

The entire methodology is repeated for each dataset. For each, it is performed for each number of clusters k, tested systematically from 4 to 10. First, for each k, a clustering is created using all the samples and this is considered the reference clustering of size k. In other words, there is a different reference sample for each number of clusters. After that, the clustering method is applied to different random subsets of the samples. The random subsets are varied from 3 samples to the $total\ number\ of\ samples$, m (see Figure 3.1). Then the following is done N times (in our case, $N = 100$): take a random subset of the samples of the appropriate size, cluster with K-means algorithm, then calculate the distance (with a CCD) to the reference sample of size k (using Equation 3.1). It is then possible to use different subsets of the samples to generate clusterings, and see how they can differ in terms of the number of genes. Clustering is completed using the (heuristic) K-means clustering implementation with the VLFeat (MATLAB) program [80]. Here two distances (with a CD) of K-means are used. In version 1 (V1), the Euclidean distance function is applied to calculate the distance between cluster objects, and for version 2 (V2), Manhattan distance function is used.

Using Equation 3.1, the distance between the reference clustering of size k (based on all the samples) and each random subset of the samples is calculated. All distance measurement values are stored in a matrix, called *SampleDisMatrix* [Table 3.2], which stores the distances between the different random subsets of the samples and the reference.

Table 3.2: SampleDisMatrix measures the distance between clusterings and the reference. The rows indicate the different number of samples (RSS) that were randomly chosen. The columns represent each iteration with that number of samples. The entries contain distance values to the reference.

	TS_1	TS_2	TS_3	...	TS_N
RSS_3	D_{31}	D_{32}	D_{33}	...	D_{3N}
RSS_4	D_{41}	D_{42}	D_{43}	...	D_{4N}
RSS_5	D_{51}	D_{52}	D_{53}	...	D_{5N}
...	...	...	...	...	...
RSS_m	D_{m1}	D_{m2}	D_{m3}	...	D_{mN}

Let Y be a subset of the samples. Let exp_Y be a function from X that maps each gene onto its expression in sample set Y, where $exp_Y(g)$ maps each gene to an $|Y|$-tuple expression vector. Further, let exp_{ALL} be the function based on all samples. Pseudocode of the method is given in Algorithm 1.

Algorithm 1: Cluster comparison

```
calculate $G_{REF}$                                          ▷ reference clustering
for each number $i$ up to $m$ do
    for each iteration $itr$ up to $N$ do
        $Y \leftarrow$ random subset of samples of size $i$
        calculate $G_Y$                                      ▷ clustering using $Y$
        calculate $D(G_Y, G_{REF})$                          ▷ using Equation 3.1
        store in sampleDistance table in row $i$ and column $itr$
    end
end
Output SampleDisMatrix
```

To calculate the CCD $D(G_Y, G_{REF})$, three techniques are possible and all three were implemented. First, the best permutation θ can be calculated with a brute force search technique [63]. In computer science, brute force is a general search technique that iterates through each possible combination of solutions to find the optimal one. Brute force is usually a time intensive computational technique. However, it guarantees to find the best match between a reference clustering and a random subset of samples by trying every possible permutation. Second, to speed up the computation time, branch-and-bound can also be applied. Branch-and-bound considers a tree of all possible combinations of candidate solutions [18, 45] and skips entire subtrees if every resulting permutation is a guaranteed to give a worse score than the best score found so far. For example, say there are two clusterings: G_1 and G_2, and each has 4 clusters: P_1, P_2, P_3, P_4 and Q_1, Q_2, Q_3, Q_4. For observing the mapping between one clustering (G_1) to another (G_2), using branch-and-bound can possibly reduce the number of permutations that need to be considered. If any combination of the clustering set G_1 starting with P_1 compared to Q_1 fails to reach optimal criteria (because their distance is already larger than the optimal found so far), then branch-and-bound skips those (all that map P_1 to Q_1) combinations to speed up the computation time (python implementation is given in the Appendix A).

Additionally, a graph bipartite matching technique can be used to speed up the computation time in a similar fashion as mentioned by Meilă [56] for CE. In graph theory, a bipartite graph is an undirected graph with a set of vertices, V, which can be divided into two independent and disjoint sets, V_1 and V_2 [9]. In a bipartite graph, there are no edges from a vertex in V_1 to one in V_1, and no edges from one in V_2 to one in V_2. For this application, a weighted bipartite graph is used where each cluster from both clusterings is a vertex, the edges are placed on the connections between clusters from one clustering only to each cluster of the other clustering, and the weights are distances between the clusters i.e. $D(P_i, Q_j)$. The goal is to find a perfect matching (the permutation θ) between the two clusterings with the minimum sum of the distances. On the graphs, the goal is to find a matching between vertices of the first set with the second that minimizes the sum of the distances. The maximal bipartite matching finds a maximal such matching in a graph, and this can be used to find a minimal matching by negating the weights. A python package — Munkres (version 1.0.12) [3] was applied which implements the Hungarian algorithm [42]. The purpose of the Hungarian algorithm is to solve the assignment problem (determining the permutation that minimizes distance) in polynomial time. The Munkres module takes as input the k by k matrix of the distances between clusters in G_1 with G_2, and returns the minimum distance through this matrix, which is $D(G_1, G_2)$.

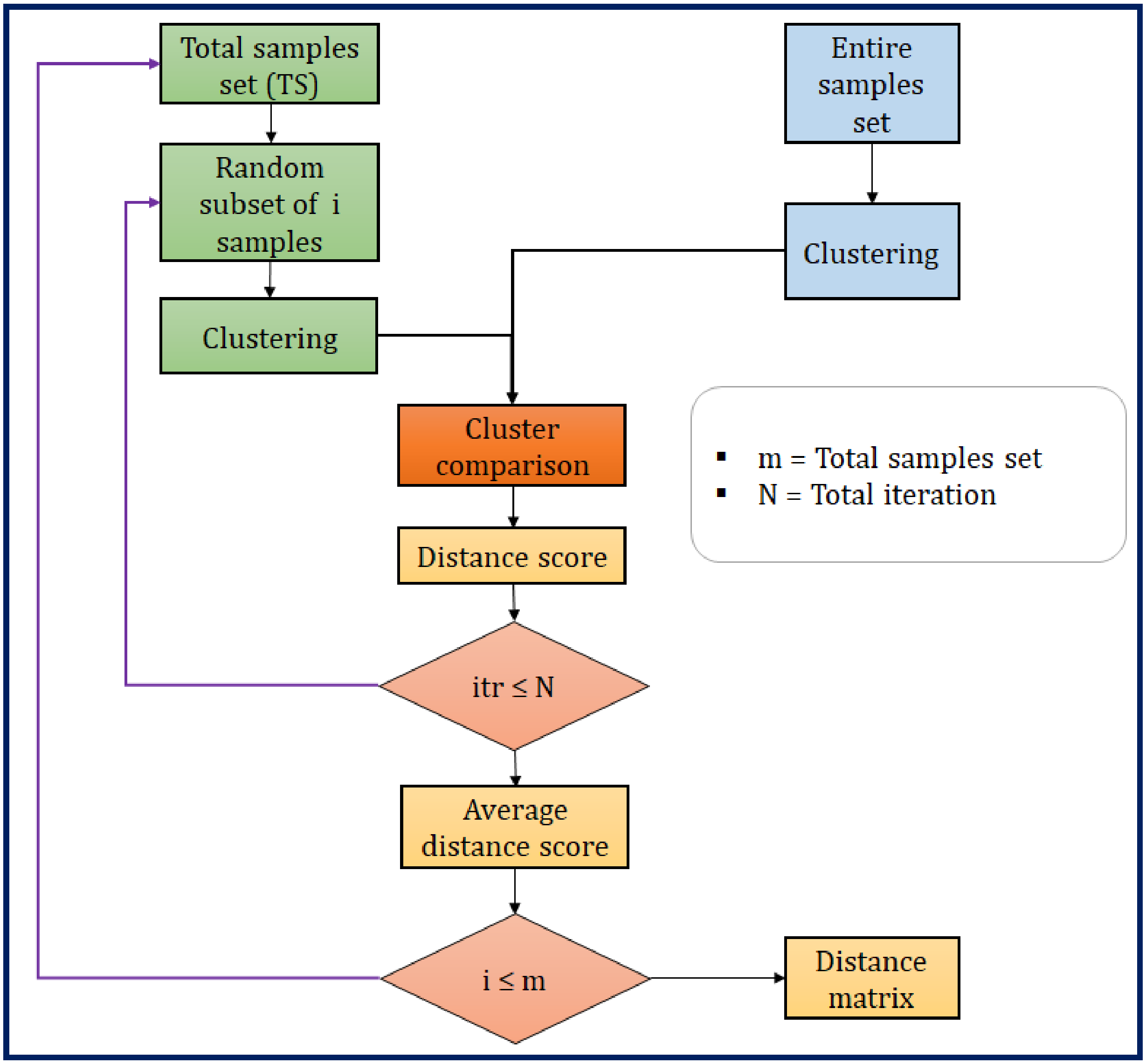


Figure 3.1: Workflow diagram of the clustering comparison technique.

3.5 Statistical analysis

Each of the three datasets can be represented an $n \times m$ matrix D, where n is the number of genes (represented as rows), and m is the number of samples (represented as columns). The mouse embryonic stem cell tissue dataset contains 78 samples, and each sample consists of 21438 genes. The *Drosophila melanogaster* dataset consists of 64 total samples with 11964 genes per sample. The *Mus musculus* multi-tissue dataset contains 70 samples (23 lung tissues, 23 liver tissues and 24 kidney tissues) and 24485 genes in each sample. To get an overall picture of the datasets, the distribution of the datasets are given in Figure 3.2, 3.3, and 3.4. All datasets have one feature vector — gene expression value. For each, the number of each gene has been plotted against (the log of) the mean expression value. It is a negatively skewed distribution due to the long left tail. The mean distribution indicates that most of the gene expression values of all samples are close to zero.

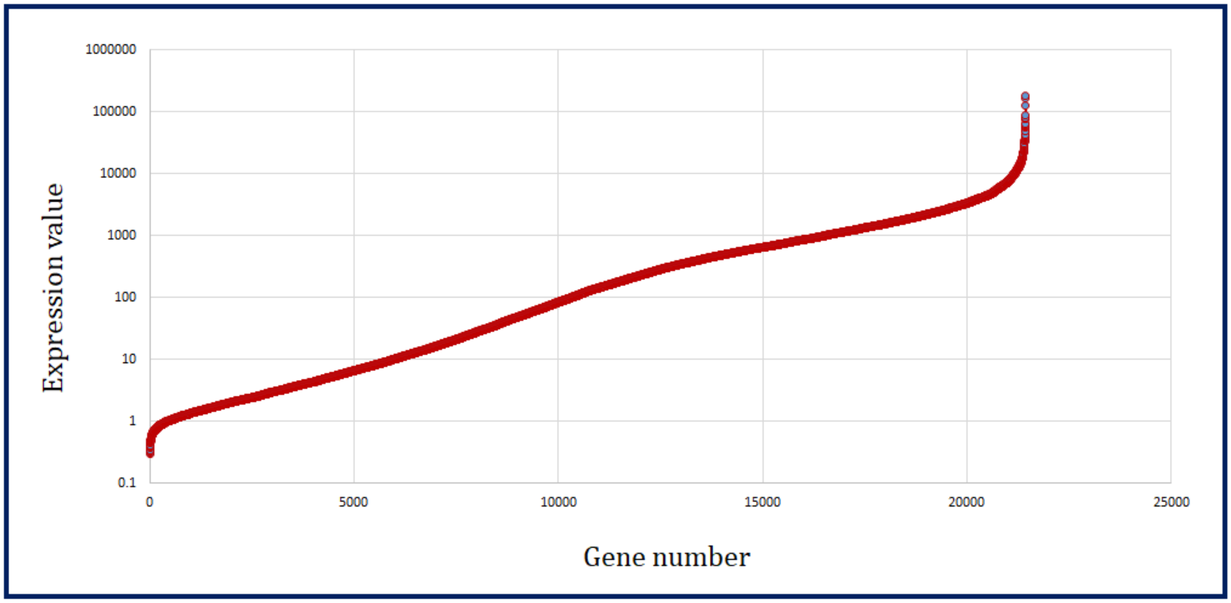


Figure 3.2: Sorted mean log distribution of stem cell tissue gene expression data.

Quantile-Quantile plot

Besides the mean sorted log distribution, another graphical data distribution technique is used: quantile-quantile plot (Q-Q plot). A Q-Q plot can help show the difference between two probability distributions [88]. A Q-Q plot compares two samples by plotting quantiles against each other. This method is used to test the samples location, skewness of distributions, and the overall pattern for specific samples. Here we compare our datasets (the mouse stem cell tissue dataset, the *Drosophila melanogaster* dataset, and the mouse multi-tissue dataset) against a Gaussian distribution. Randomly 12 samples are taken from each of the three datasets. Figure 3.5, 3.6, and 3.7 show Q-Q plots of the datasets. All the distributions are negatively skewed, which

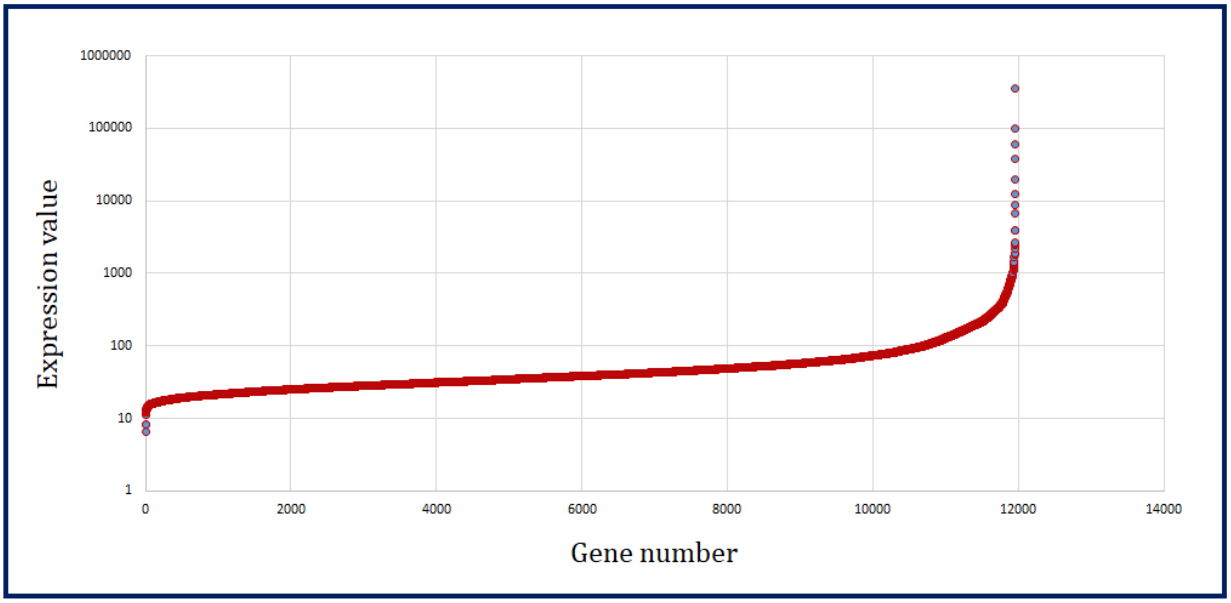


Figure 3.3: Sorted mean log distribution of *Drosophila melanogaster* multi-tissue, multiple microclimates data.

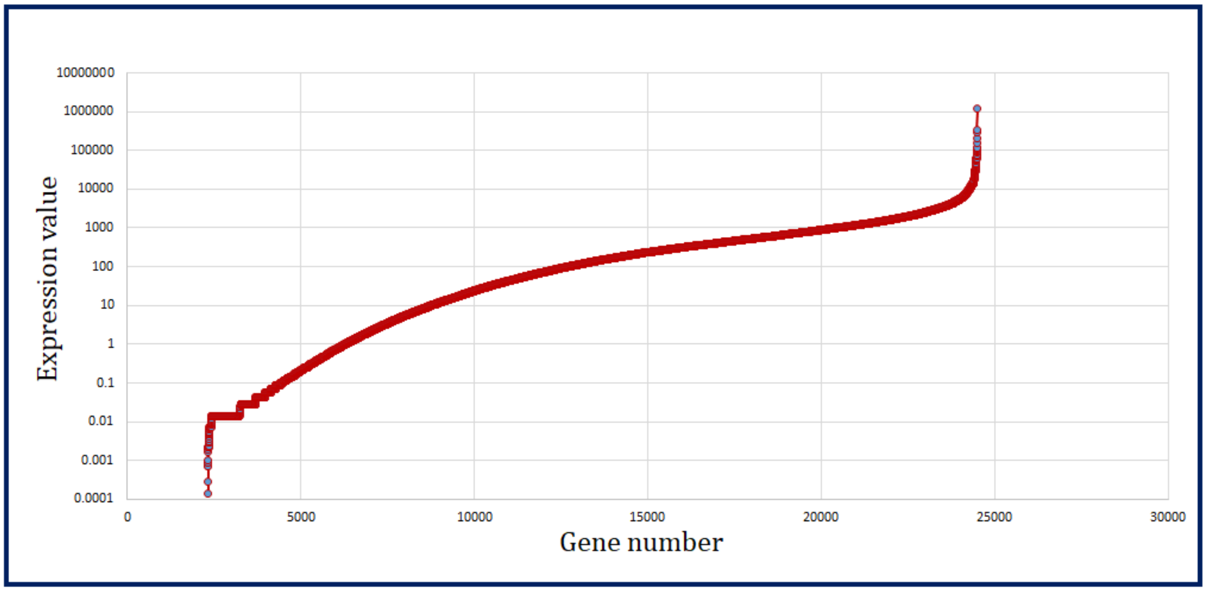


Figure 3.4: Sorted mean log distribution of *Mus musculus* multi-tissue data.

are similar to the mean sorted log distribution, supporting the negative skew previously described.

One factor ANOVA

Various statistical tools are applied to test the differences among different groups. The ANOVA (analysis of variance) is compatible with multiple groups analysis. An ANOVA tests the differences of variances between or within various conditions or groups.

Here, one factor ANOVA is used to test the effect of different numbers of clusters on the distance scores. One factor ANOVA is used to test the significance of an independent variable on a dependent variable. When one independent variable is used, then it is referred to as a one factor ANOVA test. In our case, the numbers of clusters are used as an independent variable, and the distance scores are used as dependent variables. The null hypothesis is: $H_0 = \mu_1 = \mu_2 = \mu_3 = \ldots = \mu_k$ where k is the number of groups and the mean of group i is μ_i. The alternate hypothesis (H_A) is that at least two groups mean's are significantly different.

Spearman's rank correlation coefficient

The Spearman's rank correlation coefficient is a nonparametric correlation rank statistic [49, 60]. It measures the strength of an association between two variables.

In Spearman correlation, first, all data samples need to be ranked. For example (see Table 3.3 for a toy example), if there are two samples, X and Y, they are ranked, in column 3 for sample X, and column 4 for sample Y, respectively. Differences between ranks are placed in column 5, and get all positive values, the values of the fifth column are squared in column six. Spearman's correlation is calculated using this following equation:

$$r_s = 1 - \frac{6 \sum D_i^2}{n(n^2 - 1)}, \tag{3.2}$$

where n is the total number of samples and D_i is the difference in paired ranks of gene i in the two samples.

In this toy example, Spearman's correlation coefficient is found to be 0.94. If Spearman's correlation coefficient is a value close to 1, then in Cartesian coordinates, every increase in the x-axis gives an almost equal increase in the y-axis. If Spearman's correlation coefficient is close to -1, then every increase in x, gives a decrease in y. Therefore, in this example, there is a strong positive correlation.

An R function called *cor.test* is used to implement the Spearman's rank correlation coefficient (see Section 4.2). Another R function, named *corrplot* is used here to visualize the correlation among various numbers of clusters and their distance scores.

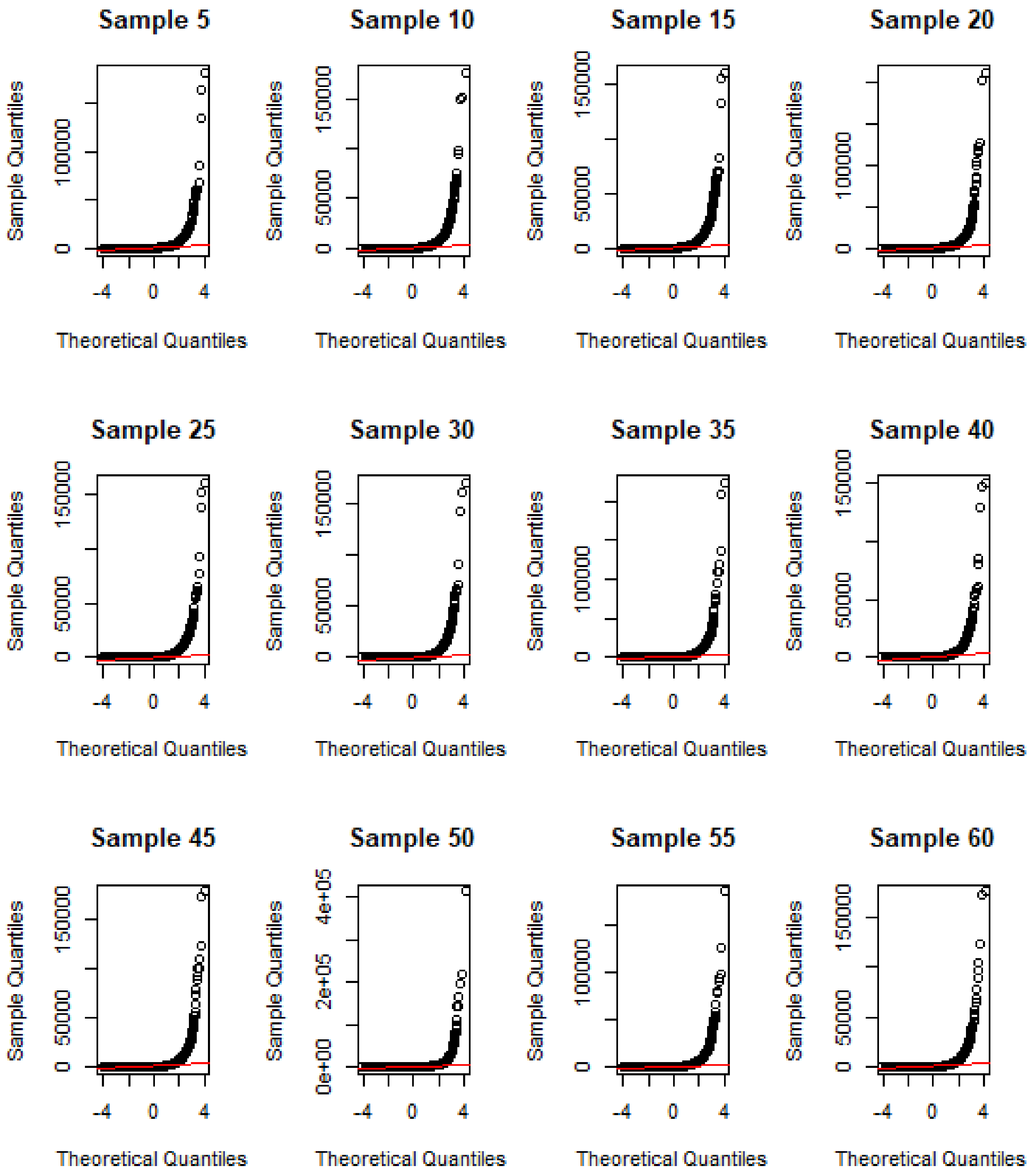


Figure 3.5: Q-Q plot of mouse embryonic stem cell tissue gene expression dataset.

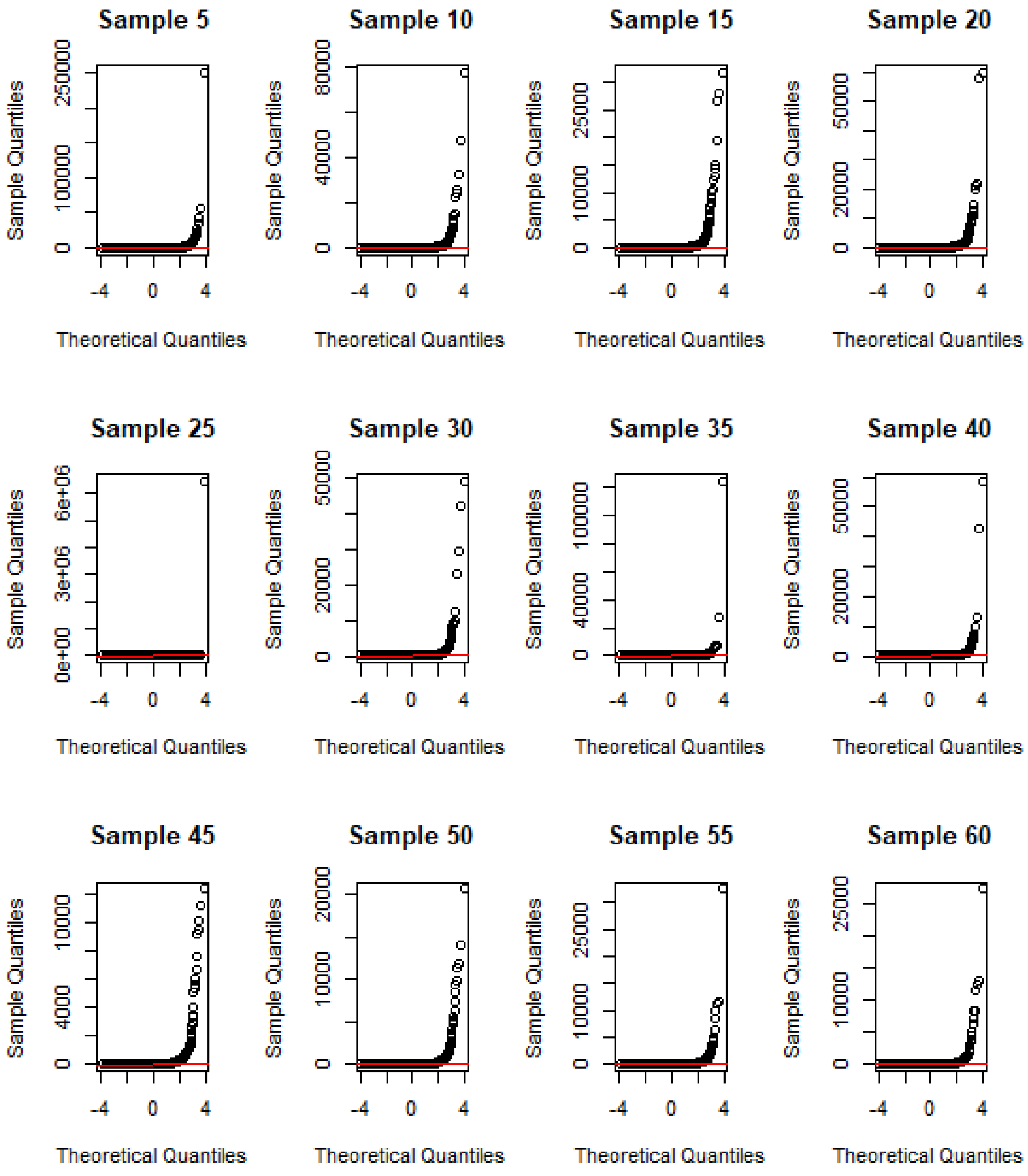


Figure 3.6: Q-Q plot of *Drosophila melanogaster* gene expression dataset.

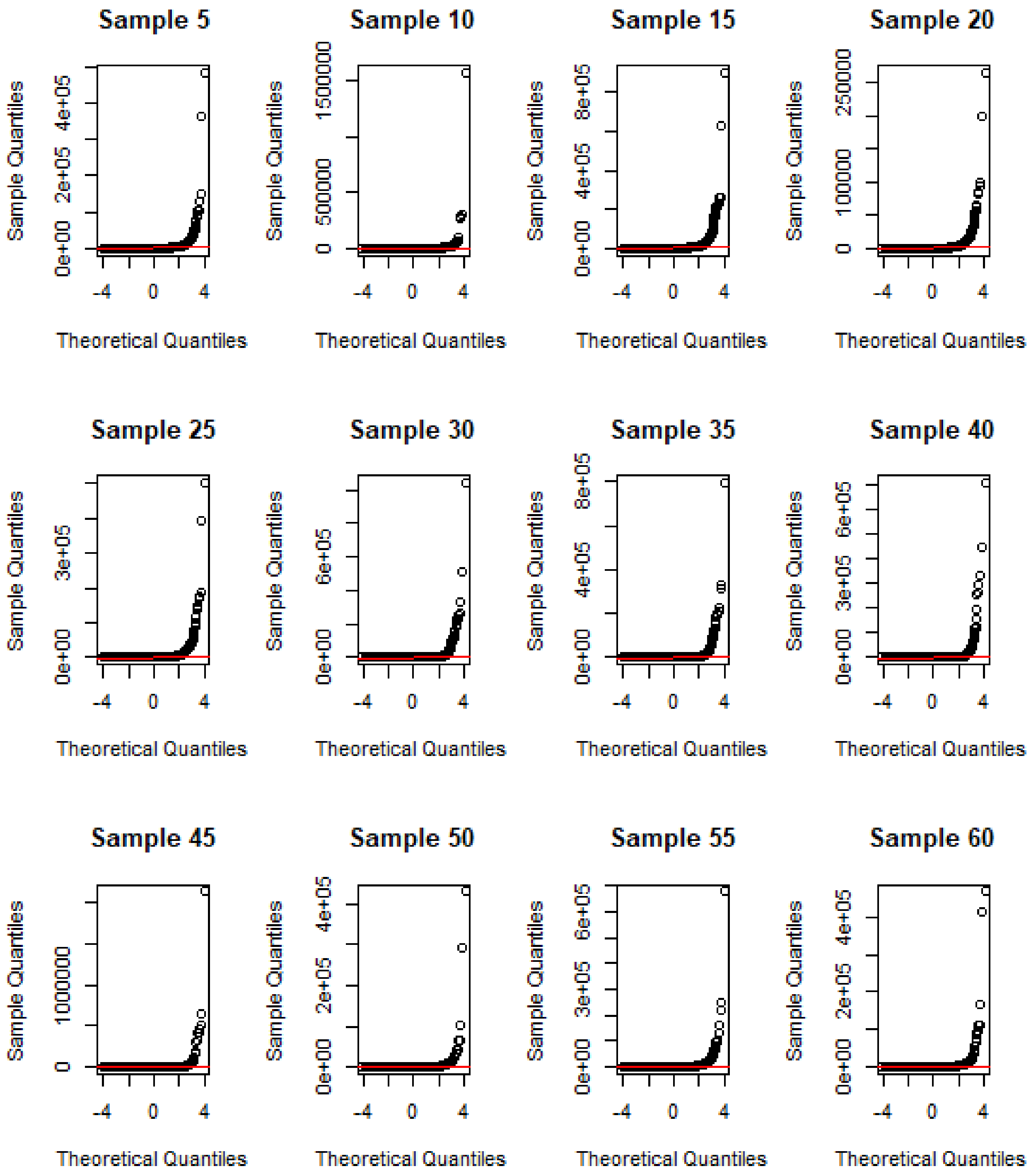


Figure 3.7: Q-Q plot of mouse multi-tissue gene expression dataset.

Table 3.3: Spearman's correlation coefficient example.

Gene name	Expression value		Ranking		$D_i = X_i - Y_i$	D_i^2
	Sample X	Sample Y	Sample X	Sample Y		
0610005C13Rik	48	20	6	7	-1	1
0610007N19Rik	34	34	7	6	1	1
0610007P14Rik	1145	1271	3	3	0	0
0610009B14Rik	9	2	8	10	-2	4
0610009B22Rik	391	230	5	5	0	0
0610009D07Rik	1600	1590	1	1	0	0
0610009E02Rik	1	4	10	8	2	4
0610009L18Rik	2	3	9	9	0	0
0610009O20Rik	1179	1488	2	2	0	0
0610010F05Rik	567	618	4	4	0	0
					$\sum_i D_i = 0$	$\sum_i D_i^2 = 10$

4 Results

As mentioned in Chapter 3, to compare the consistency of clusterings, random subsets of samples are used to cluster using the K-means algorithm, and this is compared to a clustering based on all samples. This comparison is done using the clustering comparison metric (CCD) of Equation 3.1, each number of samples is used for clustering with 100 iterations, K-means is performed with different numbers of clusters that are varied systematically, and with both (CD) Euclidean and Manhattan distance. Three RNA-Seq gene expression datasets are used: mouse embryonic stem cell tissue, mouse multi-tissue, and *Drosophila melanogaster* multi-tissue and multiple microclimates. The effect of changing sample sizes on clusterings is compared using linear regression analysis in order to better understand if results improve as the number of samples increase. Then the correlation among different numbers of clusters is tested using ANOVA. Here, one factor ANOVA is used to analyze the relationship between two different numbers of clusters based on the cluster distance scoring matrices. Additionally, Spearman's rank correlation coefficient is used to test the correlation between various numbers of clusters on clusterings.

4.1 Consistency analysis

Clustering comparison results for each of the three datasets are summarized in this section.

4.1.1 Mouse embryonic stem cell tissue dataset

For this dataset, the comparison metric (CCD) is performed after clustering with (CD) distance V1 (Euclidean distance) of the K-means algorithm giving the distance score is calculated for each number of clusters from 4 to 10 and for 100 iterations using each number of samples from 3 to 78. In Figure 4.1, the x-axis represents the size of the random subset of the samples, and the y-axis shows the average distance score over 100 iterations (compared to the reference clustering) for the random subsets of the samples.

When the number of clusters is 4 to 6, the average clustering comparison distance for each size of samples to the reference are somewhat similar; the distance scores vary from 300 to 700. This is relatively small given the large number of total genes. Higher distances are found for 7 to 10 clusters. The highest variation for the distance scores are found when the number of clusters 7, 8, and 10 (distance scores around 1400 to 1800). The number indicates the number of genes that need to be moved from each clustering to produce the reference. When the number of clusters is 4, increasing the sample size actually increases the distance scores. For 6 clusters, there is an initial decrease in score before stabilizing as with 9 clusters. All the clustering comparison results together with the standard error of the mean, presented in the parentheses, for V1 with this dataset are presented in Table 4.1. A linear regression will be presented in Section 4.2.

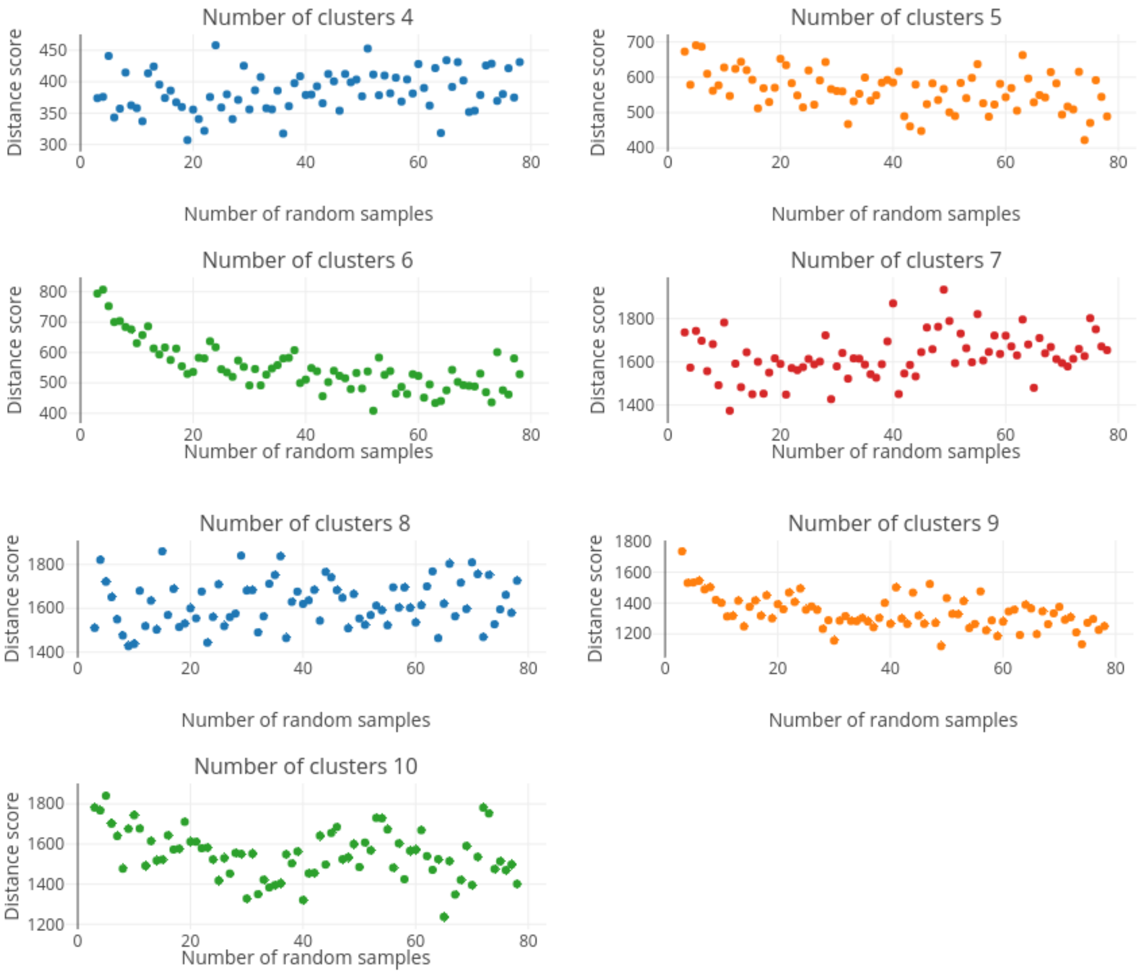


Figure 4.1: Clusterings comparison of mouse stem cell tissue dataset averaged over 100 iterations.

A similar table of results for V2 (Manhattan distance) are presented in Appendix B.1.

All results obtained with various numbers of clusters for mouse stem cell tissue and V1 are summarized in Figure 4.2. Here, the relationship between the distance scores with different numbers of clusters can be compared. The average distance scores are presented on the y-axis, and the x-axis represents the number of samples (the lines are only presented to visualize the different numbers of clusters, and do not otherwise have any significance).

Figure 4.3 shows a summary of clustering comparison using V2. When the number of clusters is 8 to 10, these show an overall higher distance score compared to the other numbers of clusters 5 to 7. The pattern

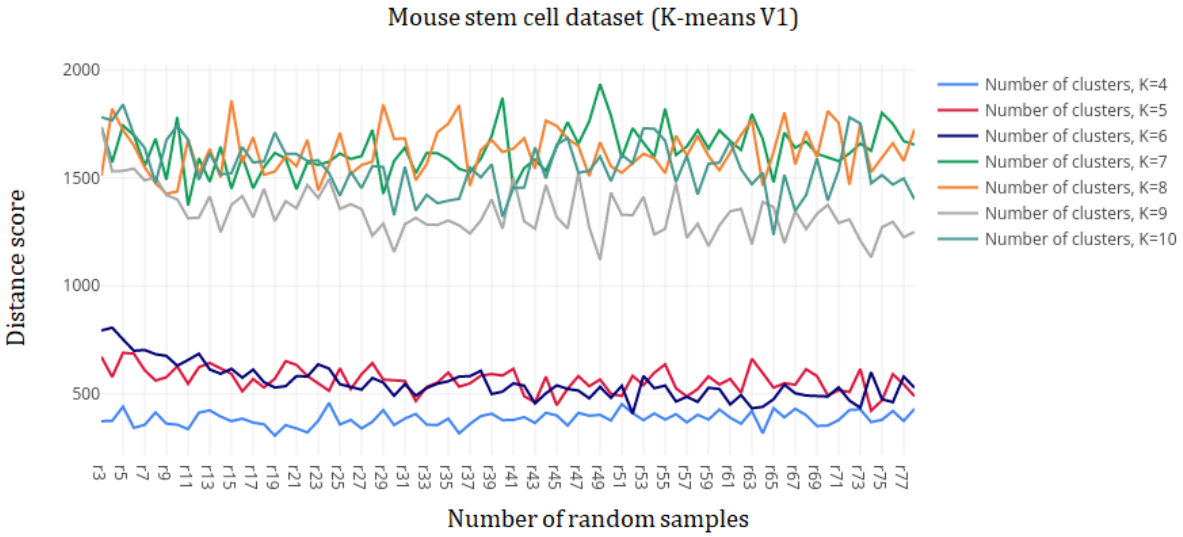


Figure 4.2: Comparison of clusterings (K-means V1) with the number of samples verses the average distance score for all numbers of clusters from 4 to 10 of the mouse stem cell dataset.

and trend of the distances scores is similar for 5 to 7 clusters. Like V1, all scores are extremely low for 4 clusters (distance scores around 500), but a decreasing pattern is more evident. For all numbers of clusters except 4 clusters, there is not much of a decreasing trend observed. In Section 4.2, linear regression will also further solidify this analysis.

Table 4.1: Average clusterings comparison over the 100 iterations, with standard error of the mean in parentheses for the mouse stem cell dataset.

Number of samples	Number of clusters 4	Number of clusters 5	Number of clusters 6	Number of clusters 7	Number of clusters 8	Number of clusters 9	Number of clusters 10
3	373.86 (51.65)	672.73 (36.76)	793.94 (36.53)	1736.39 (77.43)	1510.48 (69.85)	1734.8 (76.25)	1781.2 (78.97)
4	375.86 (53.37)	578.78 (30.48)	806.82 (48.58)	1573.1 (78.44)	1820.73 (83.65)	1530.89 (79.21)	1766.95 (92.49)
5	441.06 (45.65)	690.55 (44.26)	752.64 (40.41)	1743.03 (92.66)	1721.07 (80.41)	1533.11 (84.04)	1840.34 (90.61)
6	343.08 (54.75)	686.43 (43.94)	700.19 (41.19)	1697.63 (80.26)	1650.79 (75.79)	1543.43 (80.25)	1700.9 (93.38)

Table 4.1 continued from previous page

Number of samples	Number of clusters 4	Number of clusters 5	Number of clusters 6	Number of clusters 7	Number of clusters 8	Number of clusters 9	Number of clusters 10
7	357.14 (54.99)	609.8 (39.28)	703.64 (35.26)	1556.77 (87.06)	1549.43 (87.7)	1488.52 (78.3)	1639.28 (94.36)
8	414.59 (53.42)	562.04 (34.56)	684.01 (34.88)	1681.83 (86.91)	1475.8 (88.4)	1502.78 (90.17)	1476.81 (95.56)
9	362.4 (57.78)	577.24 (40.19)	676.24 (36.39)	1491.67 (84.24)	1427.26 (78.17)	1420.46 (86.15)	1675.39 (86.9)
10	357.98 (55.41)	627.62 (40.65)	630.7 (46.42)	1782 (84.03)	1437.24 (79.66)	1401.12 (83.93)	1743.17 (113.37)
11	337.06 (58.34)	547.15 (44.66)	657.44 (37.5)	1373.4 (77.72)	1679.9 (86.07)	1313.35 (85.44)	1676.9 (106.58)
12	413.45 (50.04)	623.92 (37.47)	686.29 (48.06)	1591.52 (90.82)	1518.92 (83.94)	1316.51 (78.79)	1491.43 (90.87)
13	424.2 (54.18)	643.78 (46.01)	613.4 (33.09)	1482.44 (94.76)	1634.89 (88.31)	1415.42 (85.08)	1614.65 (99.93)
14	395.43 (54.92)	620.16 (44.03)	593.58 (41.02)	1644.24 (87.5)	1502.71 (83.15)	1248.78 (82.33)	1517.37 (103.4)
15	374.15 (58.57)	592.89 (38.71)	616.54 (38.59)	1449.85 (90.94)	1859.56 (83.15)	1375.89 (86.62)	1522.26 (102.52)
16	385.88 (57.19)	512.17 (38.28)	575.42 (36.95)	1600.25 (89.82)	1569.48 (91.53)	1416.34 (93.12)	1643.4 (111.39)
17	367.25 (57.06)	569.04 (40.09)	612.97 (34.49)	1452.55 (83.67)	1688.4 (93.79)	1317.97 (86.69)	1572.43 (97.51)
18	359.74 (58.51)	530 (40.21)	554.8 (38.31)	1550.1 (96.01)	1514.63 (94.58)	1449.81 (88.27)	1576.41 (102.39)
19	307.04 (60.67)	570.4 (38.33)	529.67 (34.37)	1616.03 (94.43)	1530.49 (98.69)	1300.89 (84.74)	1710.52 (106.99)
20	355.61 (60.88)	652.11 (43.27)	536.05 (34.49)	1590.31 (89.79)	1599.79 (87.62)	1392.76 (86.72)	1611.49 (109.09)
21	340.59 (58.53)	634.37 (44.88)	582.72 (35.11)	1448.17 (97.3)	1554.64 (87.03)	1359.95 (82.01)	1611.09 (105.15)
22	321.85 (63.11)	583.25 (39.95)	580.68 (41.09)	1571.11 (88.05)	1676.37 (83.92)	1467.95 (84.19)	1578.82 (105.23)

Table 4.1 continued from previous page

Number of samples	Number of clusters 4	Number of clusters 5	Number of clusters 6	Number of clusters 7	Number of clusters 8	Number of clusters 9	Number of clusters 10
23	375.69 (57.64)	548.62 (40.23)	637.07 (37.11)	1561.46 (97.08)	1443.28 (79.58)	1407.31 (90.87)	1581.83 (107.31)
24	458.09 (48.99)	514.87 (40.97)	617.49 (49.68)	1575.3 (96.59)	1560.49 (88.65)	1494.41 (83.91)	1523.22 (106.25)
25	358.95 (61.23)	619.39 (45.81)	545.04 (33.79)	1613.25 (91.1)	1709.25 (96.82)	1356.89 (88.04)	1417.89 (101.51)
26	379.87 (57.07)	522.42 (41.29)	534.58 (33.52)	1588.24 (91.19)	1518.62 (84.79)	1378.52 (82.32)	1529.41 (106.12)
27	340.52 (57.74)	591.18 (43.2)	520.24 (39.35)	1601.02 (91.44)	1560.07 (77.54)	1355.98 (94.7)	1452.84 (105.52)
28	371.16 (60.82)	643.1 (46.38)	573.71 (38.39)	1722.74 (89.81)	1575.84 (86.2)	1233.06 (82.82)	1555.28 (114.79)
29	425.44 (56.05)	566.5 (45.73)	552.93 (41.72)	1427.01 (91.2)	1840.13 (99.01)	1288.84 (87.75)	1550.94 (103.14)
30	355.95 (59.46)	561.16 (41.41)	491.79 (35.73)	1578.86 (91.98)	1681.05 (90.72)	1156.81 (87.48)	1327.99 (98.38)
31	386.35 (53.97)	560.02 (38.3)	545.74 (35.7)	1640.95 (94.67)	1682.65 (94.71)	1286.12 (87.78)	1551.83 (102.27)
32	407.51 (56.39)	467.45 (34.07)	491.99 (37.65)	1522.12 (93.98)	1490.12 (88.17)	1315.52 (88.04)	1349.58 (101.52)
33	357.77 (62.17)	532.09 (45.2)	527.3 (37.21)	1615.85 (95.13)	1563.33 (82.37)	1284.29 (85.47)	1422.03 (94.51)
34	356.05 (60.18)	553.28 (41.66)	547.58 (37.84)	1614.48 (90.11)	1711.8 (94.02)	1282.88 (85.21)	1383.32 (96.07)
35	385.85 (58.49)	599.21 (45.76)	558.7 (35.86)	1587.21 (89.41)	1751.66 (95.82)	1303.05 (84.48)	1395.03 (104.21)
36	317.46 (61.2)	533.68 (43.88)	580.44 (41.63)	1542.58 (92.49)	1837.23 (92.61)	1280.66 (86.82)	1403.58 (108.42)
37	361.11 (58.83)	549.07 (41.21)	582.58 (44.52)	1526.38 (96.73)	1464.27 (93.07)	1243.11 (88.36)	1548.37 (110.35)
38	397.66 (60.88)	584.66 (44.19)	607.59 (47.22)	1588.73 (90.72)	1629.47 (93.2)	1303.33 (90.59)	1503.19 (105.15)

Table 4.1 continued from previous page

Number of samples	Number of clusters 4	Number of clusters 5	Number of clusters 6	Number of clusters 7	Number of clusters 8	Number of clusters 9	Number of clusters 10
39	408.75 (56.71)	592.45 (43.14)	499.68 (37.21)	1694.63 (93.73)	1676.08 (94.46)	1401.05 (83.62)	1562.99 (103.47)
40	378.88 (59.54)	585.26 (48.99)	511 (35.43)	1870.86 (87.32)	1620.05 (94.31)	1265.67 (88.5)	1320.67 (104.42)
41	379.56 (59.77)	616.63 (46.96)	548.94 (42.8)	1450.64 (91.2)	1636.13 (92.65)	1501.32 (81.64)	1453.69 (102.74)
42	392.65 (59.42)	489.7 (41.13)	538.36 (34.5)	1545.9 (94.19)	1683.98 (93.68)	1300.06 (86.41)	1455.89 (102.77)
43	365.69 (63.91)	461.04 (42.99)	456.32 (32.61)	1585.67 (96.39)	1543.5 (96.12)	1265.12 (90)	1641.55 (115.57)
44	412.3 (60.18)	579.4 (44.39)	502.76 (37.4)	1532.38 (90.23)	1765.71 (81.94)	1467.9 (94.97)	1497.31 (109.7)
45	400.7 (60.9)	448.03 (36.63)	539.94 (46.44)	1644.9 (94.33)	1740.99 (92.28)	1319.12 (92.6)	1654.88 (109.21)
46	353.84 (62.15)	523.9 (40.41)	523.51 (36.79)	1758.87 (98.8)	1682.58 (95.53)	1265.88 (88.61)	1684.63 (107.34)
47	412.35 (58.12)	582.78 (47.79)	515.42 (38.53)	1658.26 (97.51)	1647.9 (94.18)	1523.55 (93.55)	1524.11 (100.73)
48	399.18 (58.29)	535.66 (39.16)	479.8 (42.74)	1762.31 (91.34)	1508.67 (89.15)	1271.07 (89.06)	1533.61 (112.5)
49	403.45 (59.3)	566.84 (47.68)	532.81 (38.48)	1934.27 (82.49)	1665.08 (87.35)	1120.91 (78.6)	1599.57 (114.52)
50	376.83 (59.99)	500.94 (41.82)	481.56 (37.41)	1789.06 (86.54)	1553.69 (97.43)	1432.62 (87.43)	1485.16 (105.98)
51	452.87 (54.61)	490.31 (37.58)	537.72 (40)	1593.85 (90.64)	1524.54 (91.54)	1329.87 (86.78)	1607.41 (120.85)
52	411.48 (56.86)	584 (43.48)	408.53 (31.75)	1730.66 (100.12)	1569.13 (87.32)	1327.74 (89.3)	1568.13 (116.5)
53	378.46 (61.04)	540.95 (47.77)	583.4 (43.59)	1663.02 (97.31)	1612.2 (94.3)	1412.74 (93.28)	1729.96 (116.99)
54	409.84 (60.25)	598.27 (48.92)	526.55 (42.24)	1597.93 (93.48)	1591.15 (97.63)	1238.27 (90.6)	1727.8 (111.52)

Table 4.1 continued from previous page

Number of samples	Number of clusters 4	Number of clusters 5	Number of clusters 6	Number of clusters 7	Number of clusters 8	Number of clusters 9	Number of clusters 10
55	381.58 (56.38)	637.04 (46.75)	538.94 (34.09)	1820.89 (85.18)	1522.5 (91.7)	1264.93 (88.91)	1672.23 (113.09)
56	406.45 (56.72)	526.45 (43.77)	465 (33.97)	1606.49 (88.44)	1695.77 (103.3)	1475.67 (90.69)	1481.78 (101.47)
57	368.46 (60.54)	488.53 (43)	487.03 (38.39)	1645.92 (96.39)	1603.07 (102.72)	1223.45 (83.79)	1603.18 (110.49)
58	403.65 (59.1)	522.91 (46.95)	463.48 (40.81)	1721.78 (91.65)	1694.85 (92.6)	1287.97 (90.83)	1424.48 (104.47)
59	381.27 (62.09)	581.48 (50.02)	528.7 (42.96)	1636.51 (93.83)	1601.85 (88.97)	1185.42 (85.39)	1566.13 (105.75)
60	428.19 (57.69)	543.74 (45.16)	523.11 (37.16)	1720.96 (89.58)	1535.54 (93.98)	1279.81 (86.06)	1571.18 (116.37)
61	390.07 (59.52)	569.45 (48.28)	451.54 (43.41)	1671.19 (90.23)	1613.86 (92.56)	1345.76 (90.24)	1668.93 (115.94)
62	362.02 (62.85)	505.63 (40.88)	494.89 (39.65)	1630.1 (97.56)	1699.77 (95.22)	1357.21 (83.05)	1539.83 (103.59)
63	421.74 (59.61)	662.71 (42.23)	433.93 (37.07)	1795.92 (86.7)	1768.3 (99.11)	1193.4 (91.63)	1471.21 (103.96)
64	318.33 (61.04)	596.49 (45.82)	440.4 (37.49)	1680.32 (100.74)	1464.2 (93.63)	1389.1 (84.9)	1524.64 (104.22)
65	434.16 (56.92)	529.04 (43.95)	475.66 (36.4)	1479.56 (101.55)	1620.92 (95.34)	1364.99 (89.19)	1236.91 (104.34)
66	391.77 (59.16)	549.52 (45.54)	542.9 (40.59)	1710.14 (96.55)	1802.89 (94.04)	1198.72 (89.54)	1514.84 (106.89)
67	431.29 (54.32)	543.03 (46.32)	503.31 (37.91)	1639.51 (93.93)	1563.45 (94.91)	1346.53 (86.53)	1347.76 (106.86)
68	401.98 (60.49)	614.68 (50.24)	492.7 (44.25)	1668.74 (93.94)	1716.66 (96.71)	1263.27 (86.88)	1420.88 (103.05)
69	351.81 (63.88)	582.74 (50.37)	490.4 (35.64)	1612.49 (91.23)	1596.67 (90.64)	1333.82 (83.26)	1590.54 (115.65)
70	353.8 (58.83)	494.31 (49.13)	488.3 (35.42)	1594.62 (97)	1809.4 (102.93)	1375.69 (92.81)	1395.59 (104.76)

Table 4.1 continued from previous page

Number of samples	Number of clusters 4	Number of clusters 5	Number of clusters 6	Number of clusters 7	Number of clusters 8	Number of clusters 9	Number of clusters 10
71	378.88 (62.06)	517.15 (45.71)	530.58 (40.86)	1578.5 (96.67)	1755.6 (93.51)	1291.98 (91.36)	1535.98 (111.91)
72	425.88 (58.39)	509.11 (41.16)	469.42 (39.24)	1613.82 (92.75)	1469.16 (94.47)	1308 (85.98)	1781.09 (110.74)
73	428.74 (56.18)	615.52 (49.3)	436.13 (39.44)	1659.49 (93.35)	1752.13 (94.79)	1210.22 (85.99)	1752.2 (115.52)
74	369.59 (61.87)	422.29 (42.78)	601.13 (40.78)	1626.31 (93.52)	1527.18 (86.2)	1133.48 (91.69)	1475.39 (118.65)
75	380.31 (60.35)	470.91 (45.86)	475.79 (40.64)	1802.18 (92.73)	1595.11 (91.96)	1272.57 (89.3)	1513.61 (107.42)
76	421.45 (56.26)	591.68 (46.65)	462.01 (46.52)	1750.83 (88.49)	1661.12 (97.82)	1296.86 (88.27)	1469.84 (115.14)
77	374.83 (60.52)	544.53 (48.99)	580.7 (39.77)	1671.06 (99.46)	1579.2 (96.1)	1226.22 (85.41)	1497.77 (104.05)
78	431.31 (57.71)	489.27 (46.85)	528.78 (47.47)	1654.09 (98.86)	1726 (95.53)	1250.4 (91.47)	1400.93 (105.66)

4.1.2 Mouse multi-tissue dataset

For both V1 and V2, 100 iterations were used, and the number of clusters was varied from 4 to 10 on the mouse multi-tissue dataset. Results are summarized in Figure 4.4 and 4.5. V1 results show that lower numbers of clusters (4) have a smaller distance score than for higher numbers of clusters (7). For example, the distance score for 4 varies from 100 to 500. However, most of the distance scores for 6 and 7 vary between 2500 and 3500. For most numbers of clusters, there is an initial decrease in distance until the number of samples is around 7, at which point there is little change. For V2, a different relationship between the numbers of clusters and the distance scores was found as seen in Figure 4.5. Here, all distance scores range from of 1000 to 4000. The highest distance scores were observed for 8 clusters.

4.1.3 *Drosophila melanogaster* dataset

The results acquired from the *Drosophila melanogaster* gene expression dataset are presented next. The scatter plot (Figure 4.6) shows the clustering comparison results using a number of clusters from 4 to 10 with 100 iterations (data from Table 4.2). And finally, all numbers of clusters are summarized together in Figure 4.7.

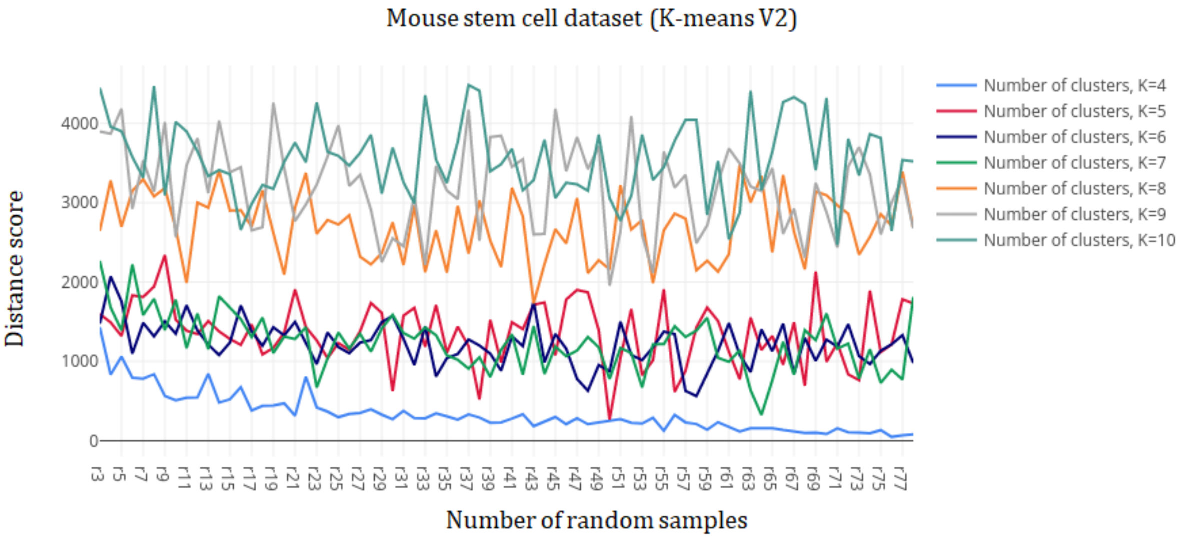


Figure 4.3: Comparison of clusterings (K-means V2) with the number of samples verses the average distance score for all numbers of clusters from 4 to 10 of the mouse stem cell dataset.

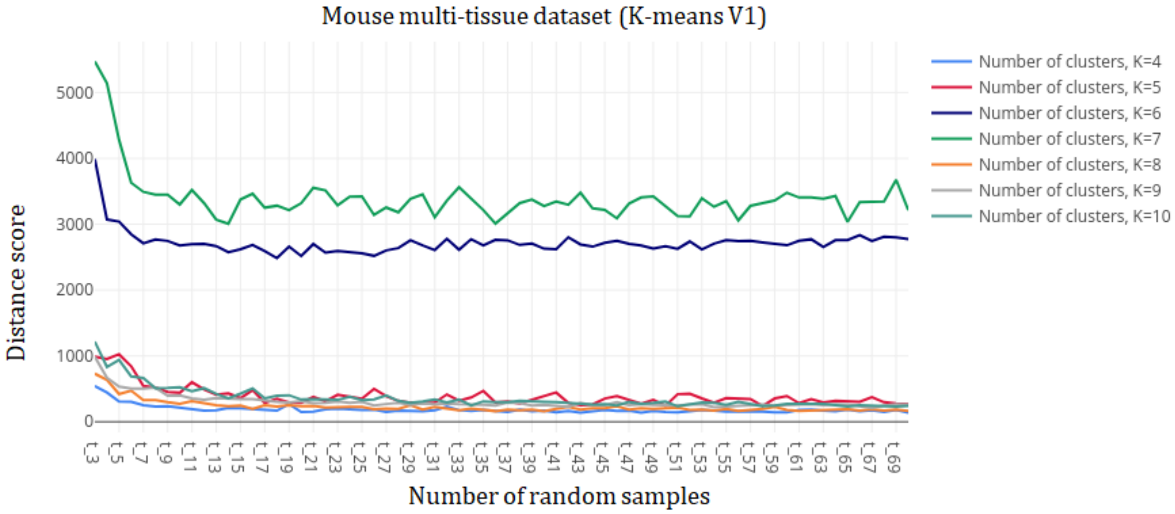


Figure 4.4: Comparison of clusterings (K-means V1) with the number of samples verses the average distance score for all numbers of clusters from 4 to 10 of the mouse multi-tissue dataset.

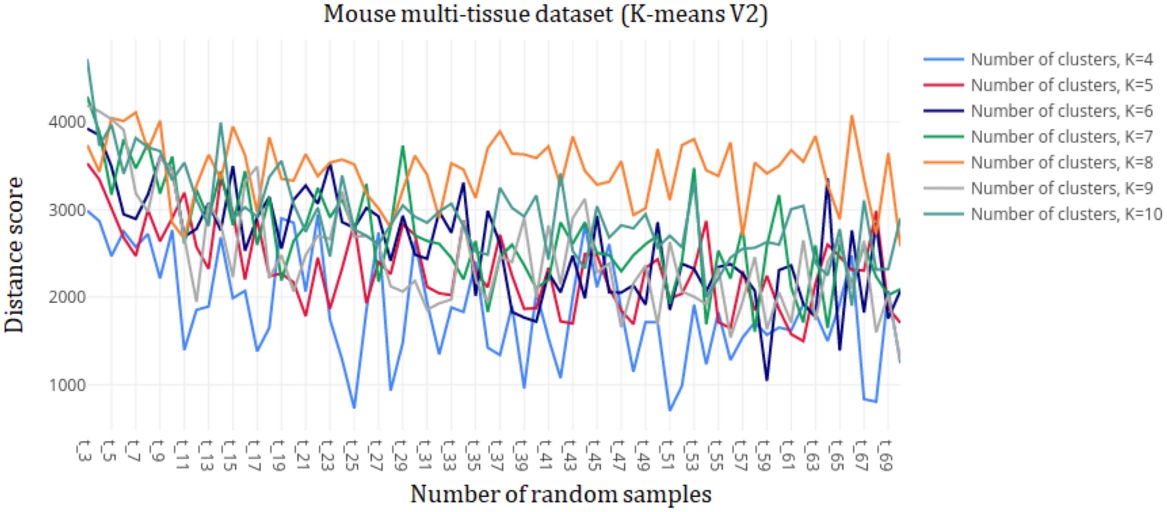


Figure 4.5: Comparison of clusterings (K-means V2) with the number of samples verses the average distance score for all numbers of clusters from 4 to 10 of the mouse multi-tissue dataset.

Similar patterns are observed in all graphs in Figure 4.6. Unlike the two mouse datasets, there is a general decreasing trend for all numbers of clusters. The reason for this discrepancy is unclear. However, the numbers themselves are very small, usually less than 50.

For V1, the graph with 5 clusters (see Figure 4.7) is an interesting exception in that it decreases far more than the others, but eventually becomes small as well when the number of samples is large. For V2 in Figure 4.8, there is again a decreasing trend until it plateaus. However, compared to V1, higher distance scores are observed for all clusters (distance scores are mostly in between 1000 to 2500 after an initial decrease).

Table 4.2: Average clusterings comparison over the 100 iterations, with standard error of the mean in parentheses for the *Drosophila melanogaster* dataset.

Number of samples	Number of clusters 4	Number of clusters 5	Number of clusters 6	Number of clusters 7	Number of clusters 8	Number of clusters 9	Number of clusters 10
3	22.75 (0.99)	370.9 (2.39)	43.76 (3.09)	63 (5.01)	85.9 (5.25)	115.48 (7.15)	145.23 (9.29)
4	20.42 (1.1)	251.4 (1.15)	38.47 (1.51)	56.31 (3.29)	66.66 (3.1)	81.53 (4.11)	102.78 (6.39)

Table 4.2 continued from previous page

Number of samples	Number of clusters 4	Number of clusters 5	Number of clusters 6	Number of clusters 7	Number of clusters 8	Number of clusters 9	Number of clusters 10
5	17.47 (1.16)	245.1 (1.03)	35.97 (1.29)	43.58 (1.82)	61.5 (3.28)	75.19 (3.25)	82.03 (3.38)
6	18.04 (0.99)	244.8 (1.18)	31.33 (1.3)	42.47 (1.76)	52.78 (1.71)	64.4 (2.28)	76.81 (2.53)
7	17.55 (1.02)	228.5 (1.11)	29.08 (1.05)	36.39 (1.43)	52.02 (1.54)	59.58 (1.9)	65.08 (1.8)
8	14 (0.86)	205.5 (1.09)	30.53 (1.32)	35.65 (1.47)	44.55 (1.62)	56.9 (2.07)	64.44 (1.87)
9	13.33 (0.75)	210.4 (1.23)	25.02 (1.27)	34.38 (1.29)	47.39 (1.8)	52.1 (1.73)	61 (1.76)
10	12.65 (0.79)	169 (0.96)	26.08 (1.23)	35.2 (1.55)	44.42 (1.3)	52.1 (1.64)	59.43 (1.81)
11	11.56 (0.79)	175.5 (0.97)	24.45 (1.19)	28.81 (1.2)	44.35 (1.48)	48.92 (1.49)	59.37 (2.14)
12	11.35 (0.81)	173 (0.93)	25.44 (1.28)	32.26 (1.36)	40.86 (1.57)	48.83 (1.73)	54.94 (2.07)
13	9.55 (0.76)	171.3 (0.94)	24.39 (1.29)	28.92 (1.39)	39.12 (1.77)	48.17 (1.68)	52.34 (1.83)
14	11.11 (0.78)	155.9 (0.89)	20.84 (1.14)	27.5 (1.25)	40.21 (1.78)	44.37 (1.67)	56.12 (1.99)
15	9.55 (0.72)	148.8 (0.93)	19.6 (1.11)	27.5 (1.28)	36.3 (1.42)	41.47 (1.58)	52.63 (1.89)
16	9.9 (0.84)	149.3 (1.1)	20.35 (1.12)	27.31 (1.38)	36.48 (1.42)	41.26 (1.39)	46.07 (1.59)
17	8.38 (0.64)	130.6 (0.85)	18.77 (1.05)	25.66 (1.14)	33.1 (1.35)	41.87 (1.5)	44.55 (1.77)
18	8.22 (0.73)	135.5 (0.91)	16.44 (0.81)	25.37 (1.4)	32.26 (1.48)	41.65 (1.83)	45.19 (1.6)
19	8.44 (0.64)	132.6 (0.86)	18.17 (1.01)	23.19 (1.32)	31.3 (1.28)	38.83 (1.57)	42.69 (1.67)
20	6.79 (0.52)	124.9 (0.84)	19.05 (1.1)	21.42 (1.25)	31.19 (1.42)	36.93 (1.38)	42.24 (1.51)

Table 4.2 continued from previous page

Number of samples	Number of clusters 4	Number of clusters 5	Number of clusters 6	Number of clusters 7	Number of clusters 8	Number of clusters 9	Number of clusters 10
21	7.18 (0.6)	115.8 (0.96)	16.45 (0.91)	19.98 (1.09)	31.33 (1.48)	36.69 (1.31)	42.38 (1.84)
22	7.26 (0.6)	101 (0.73)	15.45 (0.92)	21.61 (1.46)	28.93 (1.29)	38.04 (1.49)	38.51 (1.5)
23	6.29 (0.55)	107.6 (0.8)	17.19 (1.01)	19.74 (1.1)	29.68 (1.38)	30.93 (1.33)	38.41 (1.66)
24	5.95 (0.48)	109.1 (0.8)	14.71 (0.97)	18.8 (1.09)	27.35 (1.44)	32.75 (1.61)	38.86 (1.88)
25	6.49 (0.62)	86.7 (0.73)	12.54 (0.87)	18.99 (1.03)	27.1 (1.29)	35.56 (1.62)	34.71 (1.29)
26	5.97 (0.55)	91.4 (0.73)	14.56 (0.99)	18.63 (1.13)	28.41 (1.54)	28.91 (1.28)	36.19 (1.47)
27	6.04 (0.57)	91.4 (0.81)	13.72 (0.86)	17.71 (1.21)	24.99 (1.24)	28.41 (1.25)	36.1 (1.98)
28	4.67 (0.33)	79.9 (0.6)	12.54 (0.82)	17.02 (1.05)	25.48 (1.4)	30.61 (1.52)	33.62 (1.66)
29	4.38 (0.39)	87.2 (0.72)	10.08 (0.8)	15.96 (1.08)	25.16 (1.2)	30.66 (1.5)	30.76 (1.59)
30	5.15 (0.53)	77.7 (0.62)	11.27 (0.81)	15.52 (1)	24.15 (1.11)	28.28 (1.43)	32.31 (1.56)
31	4.15 (0.44)	72.1 (0.61)	11.98 (0.83)	14.75 (0.95)	22.64 (0.98)	26.16 (1.29)	33.48 (1.63)
32	4.1 (0.39)	75.9 (0.67)	10.48 (0.75)	15.77 (0.95)	22.42 (1.02)	27.36 (1.31)	30.95 (1.33)
33	4.72 (0.52)	61.7 (0.48)	8.55 (0.64)	12.62 (0.88)	20.97 (1.17)	27.49 (1.42)	26.68 (1.18)
34	3.9 (0.41)	61 (0.55)	9.93 (0.8)	12.04 (1.01)	18.69 (0.96)	24.58 (1.27)	28 (1.28)
35	3.46 (0.34)	55.4 (0.48)	8.39 (0.64)	11.19 (0.83)	20.36 (1.05)	26.33 (1.22)	26.29 (1.53)
36	3.64 (0.32)	57.8 (0.55)	9.21 (0.65)	12.29 (0.93)	19.71 (1.06)	22.29 (1.18)	26.34 (1.38)

Table 4.2 continued from previous page

Number of samples	Number of clusters 4	Number of clusters 5	Number of clusters 6	Number of clusters 7	Number of clusters 8	Number of clusters 9	Number of clusters 10
37	4.25 (0.52)	65.6 (0.71)	7.3 (0.55)	11.95 (0.82)	20.25 (1.05)	23.01 (0.99)	25.99 (1.4)
38	3 (0.32)	55.6 (0.55)	8.63 (0.69)	11.95 (0.86)	17.03 (0.82)	22.65 (1.15)	24.75 (1.34)
39	3.07 (0.23)	46.8 (0.4)	8.84 (0.73)	11.66 (0.95)	18.38 (1.07)	23.89 (1.27)	22.44 (1.14)
40	3.46 (0.37)	51 (0.66)	6.59 (0.64)	9.8 (0.71)	17.88 (0.94)	22.82 (1.2)	23.5 (1.29)
41	3.08 (0.3)	40.7 (0.39)	7.29 (0.63)	10 (0.79)	17.15 (0.84)	19.49 (1.03)	23.88 (1.45)
42	2.88 (0.34)	43.3 (0.4)	6.77 (0.66)	10.35 (0.82)	16.2 (0.81)	19.23 (0.88)	23.96 (1.13)
43	2.91 (0.31)	45.2 (0.42)	5.55 (0.53)	8.94 (0.68)	16.6 (0.98)	18.95 (0.93)	21.32 (1.13)
44	2.11 (0.19)	44.3 (0.47)	6.54 (0.58)	7.84 (0.59)	17.21 (0.96)	20.42 (1.03)	21.96 (1.28)
45	2.42 (0.19)	33.2 (0.39)	5.63 (0.5)	9.51 (0.82)	15.45 (0.78)	18.47 (0.94)	19.65 (1.27)
46	2.55 (0.31)	37 (0.33)	5.21 (0.46)	8.48 (0.84)	14.38 (0.94)	18.87 (0.9)	17.67 (0.98)
47	2.59 (0.24)	31.9 (0.28)	5.15 (0.53)	7.14 (0.59)	14.96 (0.98)	15.18 (0.82)	20.66 (1.32)
48	2.29 (0.18)	40.2 (0.62)	5.88 (0.67)	7.08 (0.69)	14.82 (0.99)	18.09 (0.86)	18.45 (1.04)
49	1.81 (0.19)	24.7 (0.23)	4.64 (0.47)	7.18 (0.64)	12.47 (0.66)	15.34 (0.99)	16.21 (1.04)
50	2.07 (0.3)	33.5 (0.39)	4.04 (0.42)	6.8 (0.61)	13.86 (0.78)	17.97 (1.09)	16.03 (1.01)
51	1.73 (0.21)	30.5 (0.41)	3.91 (0.46)	5.3 (0.47)	13.79 (0.68)	14.57 (0.78)	16.1 (1.16)
52	2.17 (0.27)	30.9 (0.43)	4.22 (0.42)	5.92 (0.51)	13.57 (0.76)	13.96 (0.68)	16.58 (0.93)

Table 4.2 continued from previous page

Number of samples	Number of clusters 4	Number of clusters 5	Number of clusters 6	Number of clusters 7	Number of clusters 8	Number of clusters 9	Number of clusters 10
53	1.65 (0.21)	25.9 (0.25)	3.83 (0.39)	5.17 (0.5)	12.19 (0.63)	15.47 (0.87)	15.8 (1.07)
54	1.62 (0.15)	23.9 (0.33)	3.48 (0.37)	5.61 (0.64)	13.37 (0.86)	15.55 (0.88)	15.18 (1.01)
55	1.47 (0.17)	24 (0.4)	2.79 (0.35)	5.07 (0.61)	12.67 (0.77)	13.88 (0.74)	12.39 (0.75)
56	1.52 (0.13)	17.4 (0.22)	3.71 (0.47)	5.37 (0.51)	11.51 (0.53)	15.44 (0.96)	12.56 (0.86)
57	1.26 (0.13)	17.7 (0.27)	2.96 (0.38)	4.69 (0.48)	11.73 (0.64)	12.29 (0.7)	12.4 (0.95)
58	1.6 (0.18)	17.1 (0.22)	3.33 (0.38)	4.87 (0.52)	10.88 (0.56)	12.12 (0.7)	12.05 (0.75)
59	1.48 (0.16)	14.7 (0.17)	2.55 (0.32)	3.92 (0.39)	10.29 (0.61)	12.89 (0.64)	12.55 (0.8)
60	1.06 (0.11)	14.2 (0.17)	3.26 (0.41)	4.42 (0.47)	11.83 (0.63)	13.39 (0.81)	9.43 (0.59)
61	0.97 (0.12)	15.2 (0.17)	2.28 (0.31)	3.86 (0.46)	10.69 (0.48)	11.36 (0.7)	11.35 (0.73)
62	0.98 (0.11)	12.4 (0.13)	2.3 (0.33)	3.93 (0.52)	9.73 (0.41)	11.26 (0.63)	11.06 (0.73)
63	1.31 (0.18)	19.3 (0.33)	1.79 (0.24)	3.37 (0.4)	10.04 (0.58)	11.53 (0.71)	11.65 (0.77)
64	0.81 (0.09)	13.5 (0.22)	1.91 (0.26)	2.99 (0.37)	9.53 (0.47)	11.54 (0.73)	10.68 (0.82)

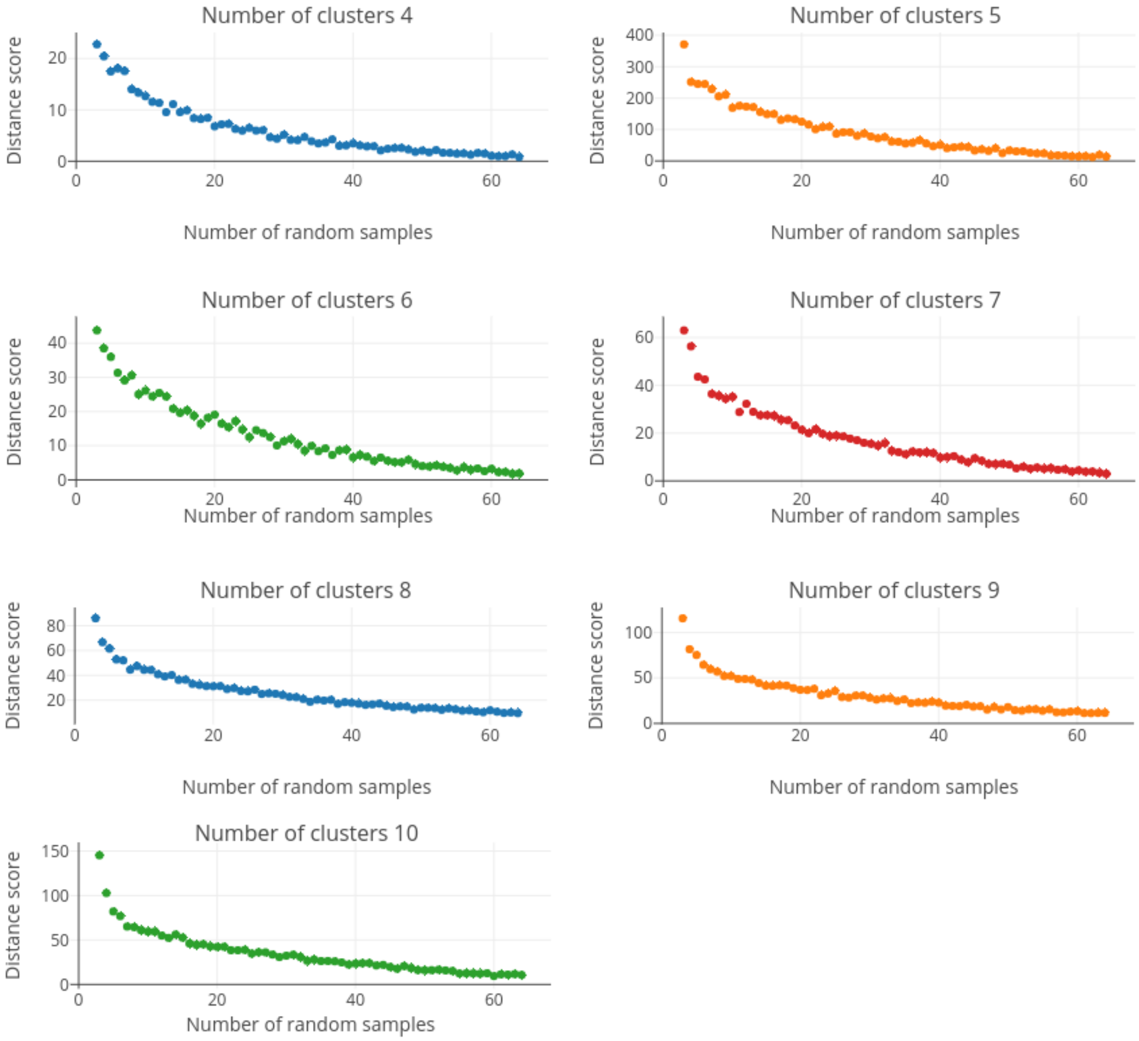


Figure 4.6: Clusterings comparison of *Drosophila melanogaster* dataset averaged over 100 iterations.

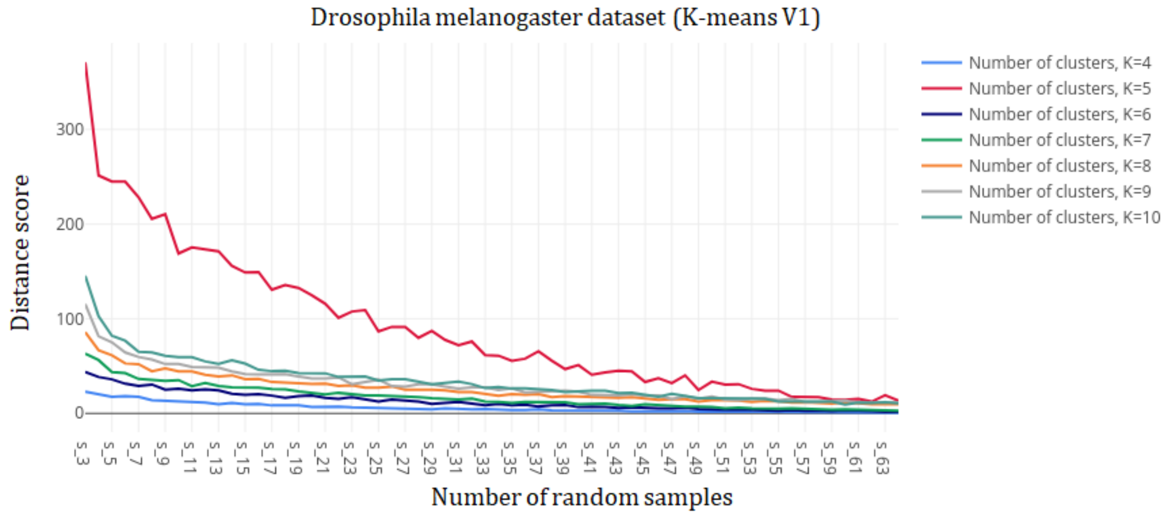


Figure 4.7: Comparison of clusterings (K-means V1) with the number of samples verses the average distance score for all numbers of clusters from 4 to 10 for the *Drosophila melanogaster* dataset.

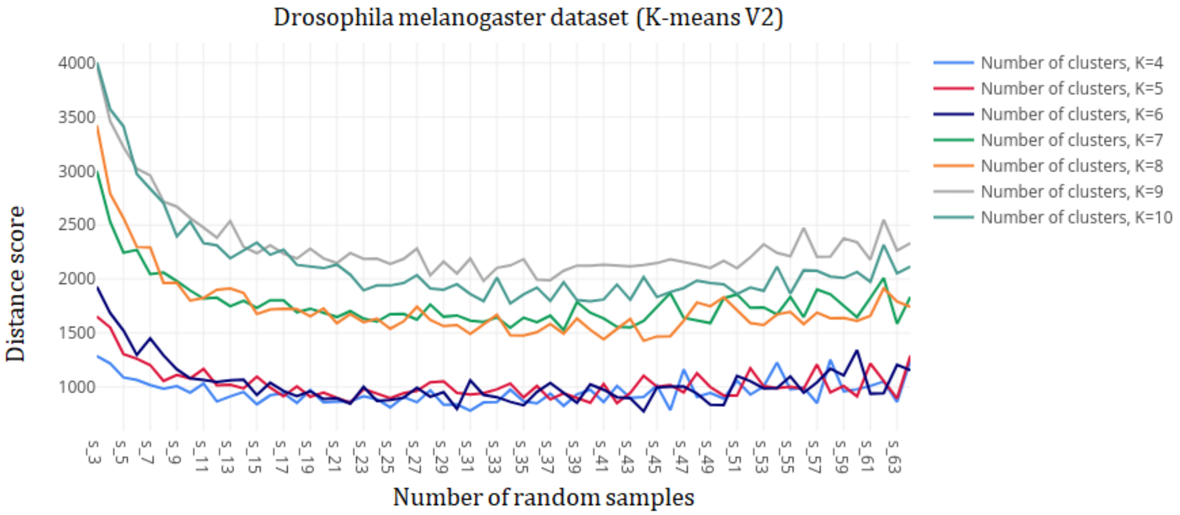


Figure 4.8: Comparison of clusterings (K-means V2) with the number of samples verses the average distance score for all numbers of clusters from 4 to 10 of *Drosophila melanogaster* dataset.

4.2 Statistical analysis

Linear regression analysis

Simple linear regression analysis was conducted to observe the trend of distance scores based on varying the number of samples. Based on the distance score results in the three datasets, Table 4.3, 4.4, and 4.5 presents the changes in distance scores for changes in random sample size resulting from the linear regression analysis. Notice that slope here refers to the change in the number of genes that have to be moved to produce the reference over the changes in sample size. The linear regressions were calculated with a separate point for each of the 100 iterations. Therefore, in say Figure 4.1, for each number of samples, there was only one point representing the average over 100 iterations for that many samples. However, for the regressions, 100 separate points were added.

Table 4.3 shows that there is indeed a negative relationship between the clustering comparison distance score and the number of random samples for both versions of *Drosophila melanogaster*, as negative coefficients (slope) are found except with 4 clusters using V2. The p-values indicates that there are significant relationships between them, except for 4 and 5 clusters using V2 (if we classify it as significant if the p-value is less than 0.05). The results from mouse stem cell tissue, (see Table 4.4) indicates that there is a score decrease with an increase in the sample size for clusters (V1). For every one unit increment in sample size (random subsets of the samples), the comparison distance score decreases (negative slope) slightly by -0.008 units. Even though it is decreasing by a small amount, the p-value ($p < 0.05$) is indicating that the distance score is negatively related with the sample size increment. A similar trend is found in version 1 for 5, 6, 9, and 10 clusters. The slope is positive for 7 and 8 clusters, but the null hypothesis for 8 clusters is not rejected due to the high p-value.

Table 4.3: Linear regression results of *Drosophila melanogaster* dataset.

	Version 1 (V1)		Version 2 (V2)	
Number of clusters	Slope	p-value	Slope	p-value
4	-0.002	0.0	0.004	0.422
5	-0.003	0.0	-0.005	0.368
6	-0.005	0.0	-0.045	<0.001
7	-0.006	0.0	-0.053	<0.001
8	-0.007	0.0	-0.091	<0.001
9	-0.009	0.0	-0.098	<0.001
10	-0.011	0.0	-0.140	<0.001

A negative relationship between distance score and sample size is observed for the mouse multi-tissue dataset with V1 (see Table 4.5). For V2, Table 4.5 shows that all clusters (4 to 10) have slopes that are

Table 4.4: Linear regression results of mouse embryonic stem cell tissue dataset.

	Version 1 (V1)		Version 2 (V2)	
Number of clusters	Slope	p-value	Slope	p-value
4	-0.008	0.003	-0.083	0.0
5	-0.011	<0.001	-0.040	<0.001
6	-0.026	<0.001	-0.065	<0.001
7	0.015	0.001	-0.077	<0.001
8	0.007	0.120	-0.024	<0.001
9	-0.025	<0.001	-0.070	<0.001
10	-0.016	0.003	-0.019	0.014

decreasing and 6 and 9 clusters show a high negative relationship between comparing scores and sample sizes.

Table 4.5: Linear regression results of mouse multi-tissue dataset.

	Version 1 (V1)		Version 2 (V2)	
Number of clusters	Slope	p-value	Slope	p-value
4	-0.002	<0.001	-0.145	<0.001
5	-0.006	<0.001	-0.123	<0.001
6	-0.001	0.044	-0.211	<0.001
7	-0.007	<0.001	-0.196	<0.001
8	-0.031	<0.001	-0.032	<0.001
9	-0.038	<0.001	-0.212	<0.001
10	-0.062	<0.001	-0.171	<0.001

One factor ANOVA test results

A one factor ANOVA was conducted to compare the effect of different numbers of clusters on the distance scores. The results for V1 are summarized in Table 4.6. The analysis of variance revealed that the effect of numbers of clusters on distance score was significant, for mouse stem cell tissue $[F(6, 525) = 2847.677, p = 0]$. The result is similar for the mouse multi-tissue dataset $[F(6, 469) = 3338.055, p = 0]$. Here, the F ratio were 2847.677 and 3338.055 for stem cell tissue and multi-tissue respectively. The degrees of freedom between and within groups were 6 and 525 (mouse stem cell tissue) and 6 and 469 (mouse multi-tissue) respectively. For the *Drosophila melanogaster* dataset, a significant effect of the numbers of clusters on the distance score was found, $[F(6, 427) = 46.858, p = 4.59695E - 44]$. Hence, the degrees of freedom between groups was 6, within groups was 427, and 46.858 was observed for the F ratio.

Table 4.6: Correlation analysis of various numbers of clusters using one factor ANOVA.

<table>
<tr><td colspan="7">Mouse embryonic stem cell tissue</td></tr>
<tr><td>Source of Variation</td><td>Sum of Squares (SS)</td><td>Degrees of freedom (DF)</td><td>Mean Square (MS)</td><td>F value</td><td>P-value</td><td>F Critical</td></tr>
<tr><td>Between Groups</td><td>145321597.9</td><td>6</td><td>24220266.32</td><td>2847.677</td><td><0.001</td><td>2.115</td></tr>
<tr><td>Within Groups</td><td>4465266.028</td><td>525</td><td>8505.268</td><td></td><td></td><td></td></tr>
<tr><td>Total</td><td>149786863.9</td><td>531</td><td></td><td></td><td></td><td></td></tr>
<tr><td colspan="7"></td></tr>
<tr><td colspan="7"><i>Drosophila melanogaster</i></td></tr>
<tr><td>Source of Variation</td><td>Sum of Squares (SS)</td><td>Degrees of freedom (DF)</td><td>Mean Square (MS)</td><td>F value</td><td>P-value</td><td>F Critical</td></tr>
<tr><td>Between Groups</td><td>294504.392</td><td>6</td><td>49084.065</td><td>46.858</td><td><0.001</td><td>2.119</td></tr>
<tr><td>Within Groups</td><td>447285.119</td><td>427</td><td>1047.506</td><td></td><td></td><td></td></tr>
<tr><td>Total</td><td>741789.512</td><td>433</td><td></td><td></td><td></td><td></td></tr>
<tr><td colspan="7"></td></tr>
<tr><td colspan="7">Mouse multi-tissue</td></tr>
<tr><td>Source of Variation</td><td>Sum of Squares (SS)</td><td>Degrees of freedom (DF)</td><td>Mean Square (MS)</td><td>F value</td><td>P-value</td><td>F Critical</td></tr>
<tr><td>Between Groups</td><td>758517243.7</td><td>6</td><td>126419540.6</td><td>3338.055</td><td><0.001</td><td>2.117</td></tr>
<tr><td>Within Groups</td><td>17762066.9</td><td>469</td><td>37872.210</td><td></td><td></td><td></td></tr>
<tr><td>Total</td><td>776279310.6</td><td>475</td><td></td><td></td><td></td><td></td></tr>
</table>

Spearman's rank correlation coefficient results

A Spearman's rank correlation coefficient was used to analyze the relationship between various numbers of clusters and distance scores (V1) from 4 clusters to 10 clusters. A correlation matrix was used here to visualize the correlation results (Figure 4.9, 4.10, and 4.11 for the three datasets) using the colour coated circles (high positive/negative correlation appears in bigger circle).

Overall, there was an insignificant correlation between the various numbers of clusters for the mouse stem cell tissue gene expression data. A positive correlation was found between 6 and 9 clusters, $\rho = 0.345$, p-value $= 0.0023$. Comparatively a positive correlation was also observed between 6 and 10 clusters. Insignificant correlation was found for 4 clusters. With the mouse multi-tissue dataset, a positive correlation was observed between 4 and 10 clusters. However, a negative correlation was found between 6 and 10 clusters, $\rho = -0.258$, p-value $= 0.033$.

A strong positive correlation pattern between different numbers of clusters was found for the *Drosophila melanogaster* dataset. Most of the pairs of numbers of clusters were positively correlated. A strong positive correlation was found between 4 to 10 clusters. For example, between 5 and 7 clusters, $\rho = 0.990$ with p-value $= <0.001$. Between 8 and 9 clusters, the p-value $= <0.001$ and $\rho = 0.990$.

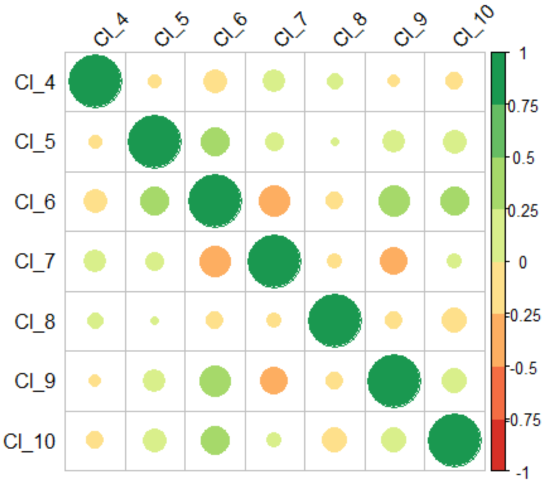


Figure 4.9: Spearman's rank correlation coefficient (calculated from V1 distance score) using various numbers of clusters for the mouse stem cell tissue dataset.

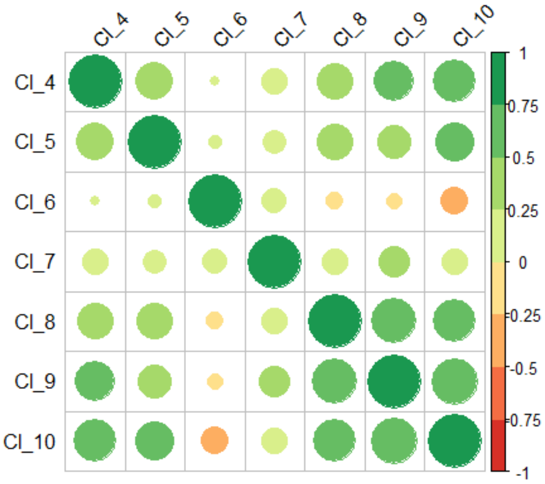


Figure 4.10: Spearman's rank correlation coefficient (calculated from V1 distance score) using various numbers of clusters for the mouse multi-tissue dataset.

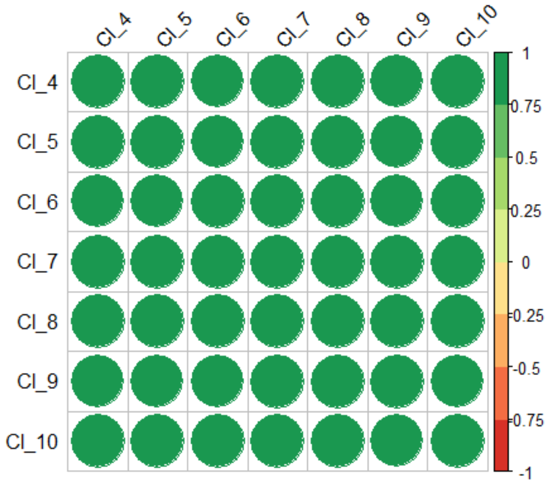


Figure 4.11: Spearman's rank correlation coefficient (calculated from V1 distance score) using various numbers of clusters for the *Drosophila melanogaster* dataset.

4.3 Execution time and comparison of algorithms

The execution time of the proposed clusterings comparison algorithms that worked successfully on our three RNA-Seq gene expression datasets were compared. The proposed algorithms were executed on a computer with an Intel Quad (40 core) CPU (Q9650) with 512 GB memory running Ubuntu 14.04.5 LTS.

From 4 to 7 clusters, the average amount of time to calculate the cluster comparison (CCD) results for 100 iterations using brute force search took less than one hour per iteration for 4 to 7 clusters. Brute force results for cluster sizes 8 to 10 appear in Figure 4.12 (one iteration). While 8 clusters is still practical, for 10 clusters, it took 9 hours, 55 minutes for *Drosophila melanogaster* and 16 hours, 45 minutes for mouse stem cell tissue (Figure 4.12). Hence, 10 clusters on the entire dataset would have taken around 50 days and 30 days for all samples of mouse stem cell tissue and *Drosophila melanogaster* data respectively.

When using the branch-and-bound mapping technique, it reduces the computation time by more than 80% for all numbers of clusters (Figure 4.13) (notice the y-axis is in hours for Figure 4.12 and minutes for Figure 4.13). They are directly compared in Figure 4.14. For 4 to 7 clusters, branch-and-bound takes only a few seconds, and for the higher numbers of clusters (e.g., 9 or 10), execution time varies from $0.5 - 2$ hours. Branch-and-bound speeds up the computation time significantly, while still keeping identical comparison scores. But it is growing exponentially, and would not be practical for a large number of iterations, or when the number of clusters is large.

However, bipartite matching takes only a few minutes for any number of clusters from 4 to 10 (while

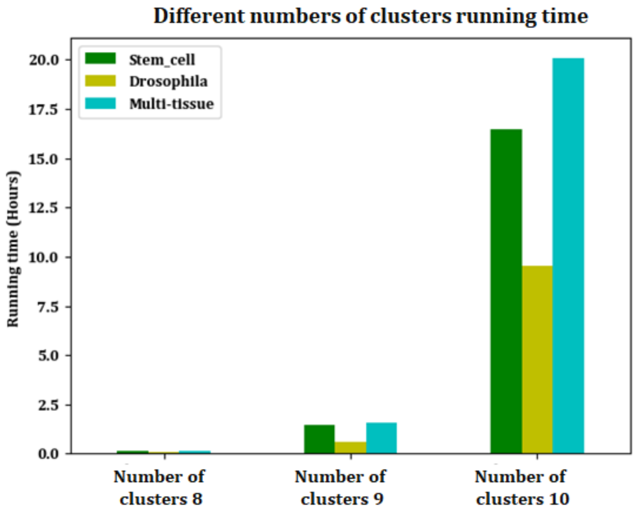


Figure 4.12: Average clusterings comparing running time for a single iteration using brute force.

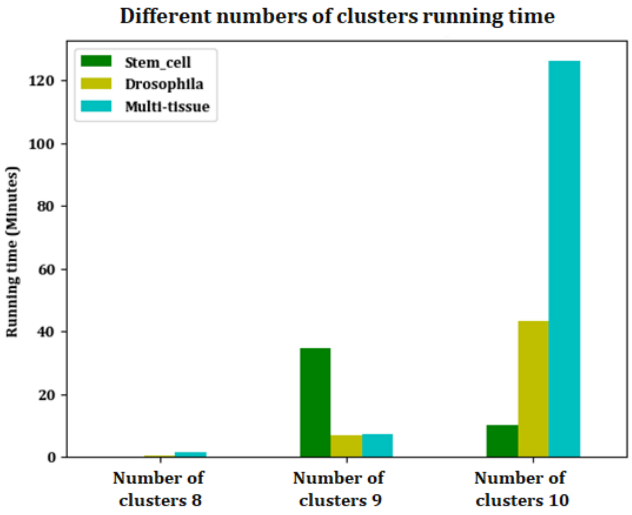


Figure 4.13: Average clusterings comparing running time for a single iteration using branch-and-bound.

achieving identical output). It is known also that it runs in polynomial time complexity and it should therefore be practical on any currently realistic dataset.

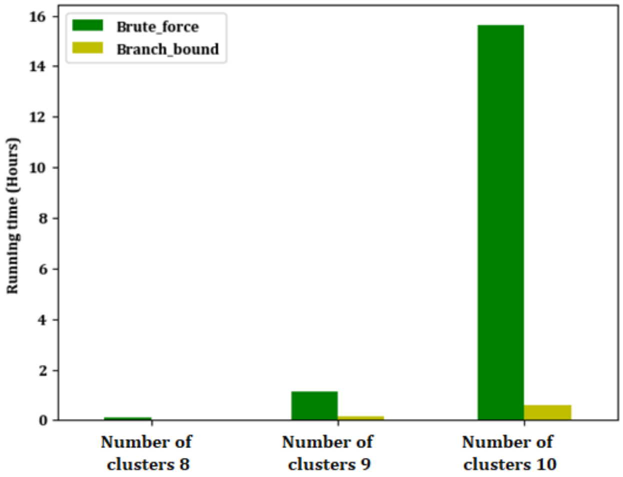


Figure 4.14: Average comparing running time between brute force and branch-and-bound using different numbers of clusters.

4.4 Discussion

Clustering can broadly be defined as grouping objects in a population based on similarity. It is one of the most popular techniques to retrieve biological information — particularly for high-throughput expression data. But, a small number of RNA-Seq samples are often used in practice, and it is desirable to know the limitations of this approach with various common tasks, such as clustering. Thus, it is important to understand the difference between clusterings built from many samples versus a smaller number of samples.

As mentioned in Section 2.3, limited research exists on the measuring of clustering comparison on RNA-Seq gene expression datasets. In reviewing the literature, no empirical results were found on the association between biological sample size, and clusterings for RNA-Seq data. The initial objective of this thesis was to find a metric for comparing clusterings and to apply it to RNA-Seq data.

Here, a clustering comparison metric was chosen specifically to help with this comparison. The proposed metric has several desirable properties. First, it is deterministic thereby always giving the same output. Also, it satisfies the properties of being a metric [56]. It is also a close variation of the existing Classification Error (CE) metric [56]. In addition, its units are in genes (or more generally, elements being clustered) which is easy to interpret rather than something unitless. This allows for intuitive distances; e.g. clustering A is different by n genes from clustering B. This also makes discussions of trade-offs between number of samples generated, and clusterings easy to understand.

The metric was then used to test the consistency of clusterings through increasing the number of random subsets of samples. For clustering algorithms, the popular (heuristic) K-means algorithm was used with both Euclidean distance and Manhattan distance. Three existing RNA-Seq gene expression datasets with a large number of samples were used: mouse stem cell tissue (78 samples), *Drosophila melanogaster* data from multiple tissues and micro-climates (64 samples), and mouse multi-tissue (70 samples). Moreover, three different algorithms, brute force search, branch-and-bound, and graph bipartite matching, were all implemented to measure the execution time of the algorithms, giving identical output. Graph bipartite matching was by far the fastest, and the only practical algorithm when there are many clusters.

In this study, one important finding is that for all three datasets, after clustering with K-means with Euclidean distance, there is usually not much decrease in distance score to the reference as the number of samples increases past same small threshold. In situations where the scores did decrease gradually, the scores themselves were small. For the mouse multi-tissue dataset, for all numbers of clusters, past approximately 7 samples, the distance scores barely change. For the mouse stem cell dataset, the distances do not decrease much (and sometimes increase). The main exception is with the *Drosophila* dataset using 5 clusters, where the distances seem to continue to decrease as the sample size increases. The actual distances values are also quite interesting. For *Drosophila*, with only a small numbers of samples, there are usually less than 50 genes different from the reference. The distances are higher for the other two datasets. It is unclear why there is such discrepancy between datasets.

For Manhattan distance, the results are somewhat different. Firstly, the distances themselves are usually higher with Manhattan distance versus the same dataset using Euclidean distance (therefore, more genes are needed to be moved to transform into the reference). Also, there is usually more variation and little separation between numbers of clusters. For the *Drosophila melanogaster* dataset, for 5, 6, and 7 clusters, the distances has an initial decrease, whereas it is more of a continual decrease with 4 clusters.

In principle, when clustering based on all samples two times, then comparing the clusterings, one would expect that the scores would be low. Certainly running a reference clustering against itself using the proposed metric produced a clustering comparison score of zero. However, after increasing to the highest number of samples, such as 70 samples for the multi-tissue dataset, the distance scores using the proposed metric for K-means (for any K tested) is often high, which means the two clusterings themselves are quite different. This is caused by the iterative K-means algorithm used which can give different output from the same input. Table 4.7 shows an example using the multi-tissue dataset where the sizes of the clusters themselves (on the same data) are varying significantly with K-means, and therefore the distance to the reference has to be large. Alternative K-means implementations such as an exact method that is deterministic might produce different results, however usually a heuristic version is used in practice.

Table 4.7: Example clusterings of multi-tissue dataset for three iterations. Each row indicates the number of an iteration of all 70 samples, and columns represent the number of genes in each cluster.

Mouse multi-tissue dataset	Number of genes in cluster 1	Number of genes in cluster 2	Number of genes in cluster 3	Number of genes in cluster 4
Reference sample	17639	5440	2	1404
Iteration 1 of Random 70 samples	17613	88	1378	5406
Iteration 2 of Random 70 samples	19524	65	2	4894
Iteration 3 of Random 70 samples	16441	5506	2033	505

For higher numbers of clusters, a similar pattern was found as with smaller numbers of clusters. Not much is known regarding the relationship between the numbers of clusters and the distance scores.

This quantitative research suggests that for the datasets tested, a lower number of samples seems to be sufficient for the purpose of K-means clustering but there are exceptions. When it did continuously decrease, the scores were small. An improved understanding of when it gradually decreases versus plateauing is unknown and is left as future work. But, the frequent plateauing or small distances in every case for Euclidean distance might influence the use of a smaller sample size for RNA-Seq data analysis in certain circumstances. However, more datasets are needed to draw any conclusions.

5 Conclusion and Future Work

The current study aimed to develop a methodology for testing the effects of RNA-Seq sample size to clusterings. For this, an appropriate metric for clustering comparison was chosen. It is simply the number of genes that need to be moved from one clustering to produce the other one. The distance unit is genes which is easy to interpret, especially in the context of understanding the trade-offs between sample size and clustering. It is also relatively time efficient to calculate the metric using the maximal bipartite matching problem.

It was used to analyse RNA-Seq gene expression clusterings. Three RNA-Seq gene expression datasets were used: mouse stem cell tissue dataset, *Drosophila melanogaster* dataset, and a mouse multi-tissue dataset. All had between 64 and 78 samples. For the clusterings, a strict partition clustering algorithm, the K-means (heuristic) algorithm was used here with both Euclidean and Manhattan distance. This part of the thesis work was undertaken in two phases. First, the methodologies for using the clustering comparison to evaluate clustering using the data is described. Then, it is applied to investigate the effects of increasing the number of RNA-Seq samples on clusterings. Additionally, the numbers of clusters was systematically varied to test the impact on the clusterings.

The distances between clusterings obtained from different random subsets of the samples versus a clustering obtained from all samples was measured as a function of the number of samples. For Euclidean distance, most often there is not much decrease in threshold past some small threshold. But there are exceptions, such as with the *Drosophila melanogaster* dataset with 5 clusters, but even in this case, distance scores were very small. Therefore, most often, after some relatively small number of samples, there was only a small amount of benefit (in terms of genes that needed to be moved in the clustering) in adding additional samples. Compared to Euclidean distance, Manhattan distance shows a different pattern. Overall, higher distance scores were found in all three datasets. In the mouse stem cell tissue and mouse multi-tissue datasets, most of the distance scores vary from 500 to 4000 (number of genes to be moved to transform the clustering into the reference) for various numbers of clusters. The *Drosophila melanogaster* shows the lowest variation of 50 to 2500.

Our comparison shows that increasing the number of samples in most cases has a limited impact on the clusterings, and when it does, the distances themselves are small. This could help to reduce the number of samples, and cost.

5.1 Limitations and future directions

There are many limitations to the current approach that should be addressed in future work. First, only K-means was tested, and only with Euclidean distance and Manhattan distance. Furthermore, the range of clusters tested was only from 4 to 10 (with 100 iterations). Moreover, it is possible that only creating one reference for each number of clusters is limiting, and it is beneficial to create more. Additionally, samples were taken randomly from the total number of samples without taking into account conditions (results might change if an equal number of samples for each condition was chosen). E.g. for mouse multi-tissue dataset, it likely makes sense to always take the same number of samples from each tissue or conditions, which was not taken into account with the random selection. Moreover, more than three datasets are needed to draw any firm conclusions or to understand why different patterns of distances versus sample size occur. Also a study of different preprocessing procedures and the impact on clustering is desired. Only RNA-Seq data was used for the analysis. Thus using the proposed measures, a future aim is to study differences between RNA-Seq and DNA microarrays. Another limitation is in the use of one strict partition clustering algorithm with multiple variations. It is desirable to use multiple overlapping and strict partition clusterings algorithm for clusterings, and present a quantitative analysis between them.

References

[1] Andrews S. FastQC: a quality control tool for high throughput sequence data. http://www.bioinformatics.babraham.ac.uk/projects/fastqc/. [Online; accessed 08-May-2018].

[2] National Human Genome Research Institute. https://www.genome.gov/sequencingcostsdata/. [Online; accessed 30-August-2018].

[3] Python Munkres Module. https://pypi.org/project/munkres/. [Online; accessed 03-October-2018].

[4] Reference genome file of Drosophila melanogaster. ftp://igenome:G3nom3s4u@ussd-ftp.illumina.com/Drosophila_melanogaster/Ensembl/BDGP5.25/Drosophila_melanogaster_Ensembl_BDGP5.25.tar.gz. [Online; accessed 25-October-2017].

[5] Ahmed N Albatineh, Magdalena Niewiadomska-Bugaj, and Daniel Mihalko. On similarity indices and correction for chance agreement. *Journal of Classification*, 23(2):301–313, 2006.

[6] Daniel Aloise, Amit Deshpande, Pierre Hansen, and Preyas Popat. NP-hardness of Euclidean sum-of-squares clustering. *Machine Learning*, 75(2):245–248, 2009.

[7] Ethem Alpaydin. *Introduction to Machine Learning*. The MIT Press, 2009.

[8] Simon Anders, Paul Theodor Pyl, and Wolfgang Huber. HTSeq—a Python framework to work with high-throughput sequencing data. *Bioinformatics*, 31(2):166–169, 2015.

[9] Armen S Asratian, Tristan MJ Denley, and Roland Häggkvist. *Bipartite Graphs and Their Applications*, volume 131. Cambridge University Press, 1998.

[10] Eric Backer and Anil K Jain. A clustering performance measure based on fuzzy set decomposition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (1):66–75, 1981.

[11] Eric Bae, James Bailey, and Guozhu Dong. A clustering comparison measure using density profiles and its application to the discovery of alternate clusterings. *Data Mining and Knowledge Discovery*, 21(3):427–471, 2010.

[12] Amit Bagga and Breck Baldwin. Entity-based cross-document coreferencing using the vector space model. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics-Volume 1*, pages 79–85. Association for Computational Linguistics, 1998.

[13] Frank B Baker and Lawrence J Hubert. Measuring the power of hierarchical cluster analysis. *Journal of the American Statistical Association*, 70(349):31–38, 1975.

[14] Paul E Black. Manhattan distance. *Dictionary of Algorithms and Data Structures*, 18:2012, 2006.

[15] Anthony M Bolger, Marc Lohse, and Bjoern Usadel. Trimmomatic: a flexible trimmer for Illumina sequence data. *Bioinformatics*, 30(15):2114–2120, 2014.

[16] Nicolas Bourbaki. *Topological Vector Spaces: Chapters 1–5*. Springer Science & Business Media, 2013.

[17] Tadeusz Caliński and Jerzy Harabasz. A dendrite method for cluster analysis. *Communications in Statistics-Theory and Methods*, 3(1):1–27, 1974.

[18] Jens Clausen. Branch and bound algorithms-principles and examples. *Department of Computer Science, University of Copenhagen*, pages 1–30, 1999.

[19] Scott Cohen and Leonidas Guibas. The Earth Mover's Distance: Lower Bounds and Invariance under Translation. Technical Report CS-TR-97-1597, Stanford University, Department of Computer Science, 1997.

[20] Thomas M Cover and Joy A Thomas. *Elements Of Information Theory.* John Wiley & Sons, 2012.

[21] David L Davies and Donald W Bouldin. A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (2):224–227, 1979.

[22] Michel Marie Deza and Elena Deza. *Encyclopedia of Distances.* Springer, 2014.

[23] Stijn Dongen. Performance Criteria for Graph Clustering and Markov Cluster Experiments. Technical report, Amsterdam, The Netherlands, 2000.

[24] Joel T Dudley and Atul J Butte. A quick guide for developing effective bioinformatics programming skills. *PLoS Computational Biology*, 5(12):e1000589, 2009.

[25] Joseph C Dunn. Well-separated clusters and optimal fuzzy partitions. *Journal of Cybernetics*, 4(1):95–104, 1974.

[26] Gary Felsenfeld and H Todd Miles. The physical and chemical properties of nucleic acids. *Annual Review of Biochemistry*, 36(1):407–448, 1967.

[27] Edward B Fowlkes and Colin L Mallows. A method for comparing two hierarchical clusterings. *Journal of the American Statistical Association*, 78(383):553–569, 1983.

[28] Sara Goodwin, John D McPherson, and W Richard McCombie. Coming of age: ten years of next-generation sequencing technologies. *Nature Reviews Genetics*, 17(6):333–351, 2016.

[29] Malachi Griffith, Jason R Walker, Nicholas C Spies, Benjamin J Ainscough, and Obi L Griffith. Informatics for RNA sequencing: a web resource for analysis on the cloud. *PLoS Computational Biology*, 11(8):e1004393, 2015.

[30] Maria Halkidi and Michalis Vazirgiannis. Clustering validity assessment: Finding the optimal partitioning of a data set. In *Data Mining, 2001. ICDM 2001, Proceedings IEEE International Conference on*, pages 187–194. IEEE, 2001.

[31] Richard W Hamming. Error detecting and error correcting codes. *Bell Labs Technical Journal*, 29(2):147–160, 1950.

[32] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. Overview of supervised learning. In *The Elements of Statistical Learning*, pages 9–41. Springer, 2009.

[33] Christian Hennig, Marina Meila, Fionn Murtagh, and Roberto Rocci. *Handbook of Cluster Analysis.* CRC Press, 2015.

[34] Lawrence J Hubert and Joel R Levin. A general statistical framework for assessing categorical clustering in free recall. *Psychological Bulletin*, 83(6):1072, 1976.

[35] Lawrence Hunter. *Artificial Intelligence and Molecular Biology.* MIT Press, 1993.

[36] Jennifer W Israel, Grace A Chappell, Jeremy M Simon, Sebastian Pott, Alexias Safi, Lauren Lewis, Paul Cotney, Hala S Boulos, Wanda Bodnar, Jason D Lieb, et al. Tissue-and strain-specific effects of a genotoxic carcinogen 1, 3-butadiene on chromatin and transcription. *Mammalian Genome*, 29(1-2):153–167, 2018.

[37] Anil K Jain, Robert P. W. Duin, and Jianchang Mao. Statistical pattern recognition: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1):4–37, 2000.

[38] Junhyong Kim. Computers are from mars, organisms are from venus. *Computer*, 35(7):25–32, 2002.

[39] Manesh Kokare, BN Chatterji, and PK Biswas. Comparison of similarity metrics for texture image retrieval. In *TENCON 2003. Conference on Convergent Technologies for the Asia-Pacific Region*, volume 2, pages 571–575. IEEE, 2003.

[40] Eija Korpelainen, Jarno Tuimala, Panu Somervuo, Mikael Huss, and Garry Wong. *RNA-Seq Data Analysis: A Practical Approach*. CRC Press, 2014.

[41] Hans-Peter Kriegel, Erich Schubert, and Arthur Zimek. The (black) art of runtime evaluation: Are we comparing algorithms or implementations? *Knowledge and Information Systems*, 52(2):341–378, 2017.

[42] Harold W Kuhn. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1-2):83–97, 1955.

[43] Solomon Kullback and Richard A Leibler. On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86, 1951.

[44] Godfrey N Lance and William T Williams. Computer programs for hierarchical polythetic classification ("similarity analyses"). *The Computer Journal*, 9(1):60–64, 1966.

[45] Ailsa H Land and Alison G Doig. An automatic method of solving discrete programming problems. *Econometrica: Journal of the Econometric Society*, pages 497–520, 1960.

[46] Ben Langmead, Cole Trapnell, Mihai Pop, and Steven L Salzberg. Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. *Genome Biology*, 10(3):R25, 2009.

[47] Bjornar Larsen and Chinatsu Aone. Fast and effective text mining using linear-time document clustering. In *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 16–22. ACM, 1999.

[48] Yingmei Lavin, Rajesh Batra, and Lambertus Hesselink. Feature comparisons of vector fields using earth mover's distance. In *Visualization' 98. Proceedings*, pages 103–109. IEEE, 1998.

[49] Ann Lehman, Norm O'Rourke, Larry Hatcher, and Edward Stepanski. JMP for basic univariate and multivariate statistics. *SAS Institute Inc., Cary, NC*, page 481, 2005.

[50] Peipei Li, Yongjun Piao, Ho Sun Shon, and Keun Ho Ryu. Comparing the normalization methods for the differential analysis of Illumina high-throughput RNA-Seq data. *BMC Bioinformatics*, 16(1):347, 2015.

[51] Kian Huat Lim and William Guy Fairbrother. Spliceman—a computational web server that predicts sequence variations in pre-mRNA splicing. *Bioinformatics*, 28(7):1031–1032, 2012.

[52] Stuart Lloyd. Least squares quantization in PCM. *IEEE Transactions on Information Theory*, 28(2):129–137, 1982.

[53] Marcel Martin. Cutadapt removes adapter sequences from high-throughput sequencing reads. *EMBnet. Journal*, 17(1):pp–10, 2011.

[54] Ujjwal Maulik and Sanghamitra Bandyopadhyay. Performance evaluation of some clustering algorithms and validity indices. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(12):1650–1654, 2002.

[55] Marina Meilă. Comparing clusterings by the variation of information. In *Learning Theory and Kernel Machines*, pages 173–187. Springer, 2003.

[56] Marina Meilă. Comparing clusterings: an axiomatic view. In *Proceedings of the 22nd International Conference on Machine Learning*, pages 577–584. ACM, 2005.

[57] Marina Meilă. Comparing clusterings — an information based distance. *Journal of Multivariate Analysis*, 98(5):873–895, 2007.

[58] Stefano Monti, Pablo Tamayo, Jill Mesirov, and Todd Golub. Consensus clustering: a resampling-based method for class discovery and visualization of gene expression microarray data. *Machine Learning*, 52(1):91–118, 2003.

[59] Davoud Moulavi, Pablo A Jaskowiak, Ricardo JGB Campello, Arthur Zimek, and Jörg Sander. Density-based clustering validation. In *Proceedings of the 2014 SIAM International Conference on Data Mining*, pages 839–847. SIAM, 2014.

[60] Kevin P Murphy. *Machine Learning: A Probabilistic Perspective.* MIT Press, 2012.

[61] Michał J Okoniewski and Crispin J Miller. Hybridization interactions between probesets in short oligo microarrays lead to spurious correlations. *BMC Bioinformatics*, 7(1):276, 2006.

[62] Katie Ovens. Integrating biclustering techniques with *de novo* gene regulatory network discovery using RNA-Seq from skeletal tissues. Master's thesis, Department of Computer Science, University of Saskatchewan, 2016.

[63] Christof Paar and Jan Pelzl. *Understanding Cryptography: A Textbook For Students And Practitioners.* Springer Science & Business Media, 2009.

[64] Karl Pearson. On the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 50(302):157–175, 1900.

[65] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[66] Jonathan Pevsner. *Bioinformatics and Functional Genomics.* John Wiley & Sons, 2015.

[67] John Quackenbush. Computational genetics: computational analysis of microarray data. *Nature Reviews Genetics*, 2(6):418, 2001.

[68] William M Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 66(336):846–850, 1971.

[69] Mark D Robinson and Alicia Oshlack. A scaling normalization method for differential expression analysis of RNA-seq data. *Genome Biology*, 11(3):R25, 2010.

[70] Andrew Rosenberg and Julia Hirschberg. V-measure: A conditional entropy-based external cluster evaluation measure. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, 2007.

[71] Peter J Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53–65, 1987.

[72] Yossi Rubner, Carlo Tomasi, and Leonidas J Guibas. A metric for distributions with applications to image databases. In *Computer Vision, 1998. Sixth International Conference on*, pages 59–66. IEEE, 1998.

[73] Erich Schubert, Alexander Koos, Tobias Emrich, Andreas Züfle, Klaus Arthur Schmid, and Arthur Zimek. A framework for clustering uncertain data. *Proceedings of the VLDB Endowment*, 8(12):1976–1979, 2015.

[74] Robert George Douglas Steel and James Hiram Torrie. *Principles And Procedures Of Statistics: With Special Reference to The Biological Sciences.* McGraw-Hill, 1960.

[75] Douglas Steinley. Properties of the Hubert-Arable Adjusted Rand Index. *Psychological Methods*, 9(3):386, 2004.

[76] Robert Tibshirani, Guenther Walther, and Trevor Hastie. Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(2):411–423, 2001.

[77] Cole Trapnell, Lior Pachter, and Steven L Salzberg. TopHat: discovering splice junctions with RNA-Seq. *Bioinformatics*, 25(9):1105–1111, 2009.

[78] Cole Trapnell, Adam Roberts, Loyal Goff, Geo Pertea, Daehwan Kim, David R Kelley, Harold Pimentel, Steven L Salzberg, John L Rinn, and Lior Pachter. Differential gene and transcript expression analysis of RNA-Seq experiments with tophat and cufflinks. *Nature Protocols*, 7(3):562, 2012.

[79] Laura J Van't Veer, Hongyue Dai, Marc J Van De Vijver, Yudong D He, Augustinus AM Hart, Mao Mao, Hans L Peterse, Karin Van Der Kooy, Matthew J Marton, Anke T Witteveen, et al. Gene expression profiling predicts clinical outcome of breast cancer. *Nature*, 415(6871):530, 2002.

[80] A. Vedaldi and B. Fulkerson. VLFeat: An open and portable library of computer vision algorithms. `http://www.vlfeat.org/`, 2008.

[81] Nguyen Xuan Vinh, Julien Epps, and James Bailey. Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. *Journal of Machine Learning Research*, 11(Oct):2837–2854, 2010.

[82] Silke Wagner and Dorothea Wagner. *Comparing Clusterings: An Overview.* Universität Karlsruhe, Fakultät für Informatik Karlsruhe, 2007.

[83] Xiaojun Wan. A novel document similarity measure based on earth mover's distance. *Information Sciences*, 177(18):3718–3730, 2007.

[84] Zhong Wang, Mark Gerstein, and Michael Snyder. RNA-Seq: a revolutionary tool for transcriptomics. *Nature Reviews Genetics*, 10(1):57–63, 2009.

[85] Matthijs J Warrens. On similarity coefficients for $2\times$ 2 tables and correction for chance. *Psychometrika*, 73(3):487, 2008.

[86] Satosi Watanabe. *Pattern Recognition: Human and Mechanical.* John Wiley & Sons, Inc., 1985.

[87] James G Wetmur and Norman Davidson. Kinetics of renaturation of DNA. *Journal of Molecular Biology*, 31(3):349–370, 1968.

[88] Martin B Wilk and Ram Gnanadesikan. Probability plotting methods for the analysis for the analysis of data. *Biometrika*, 55(1):1–17, 1968.

[89] David H Wolpert and William G Macready. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1(1):67–82, 1997.

[90] Arielle L Yablonovitch, Jeremy Fu, Kexin Li, Simpla Mahato, Lin Kang, Eugenia Rashkovetsky, Abraham B Korol, Hua Tang, Pawel Michalak, Andrew C Zelhof, et al. Regulation of gene expression and RNA editing in Drosophila adapting to divergent microclimates. *Nature Communications*, 8(1):1570, 2017.

[91] Ka Yee Yeung, Mario Medvedovic, and Roger E Bumgarner. Clustering gene-expression data with repeated measurements. *Genome Biology*, 4(5):R34, 2003.

[92] Ying Zhao and George Karypis. Evaluation of hierarchical clustering algorithms for document datasets. In *Proceedings of the Eleventh International Conference on Information and Knowledge Management*, pages 515–524. ACM, 2002.

[93] Ding Zhou, Jia Li, and Hongyuan Zha. A new mallows distance based metric for comparing clusterings. In *Proceedings of the 22nd International Conference on Machine Learning*, pages 1028–1035. ACM, 2005.

[94] Marketa Zvelebil and Jeremy Baum. *Understanding Bioinformatics.* Garland Science, 2007.

Appendix A

A.1 Python implementation

A sample implementation of clusterings comparison is given below. The code is written in python.

```python
import numpy as np
import pandas as pd
...........................
...........................
...........................

#==================================================
# Brute force
#==================================================

'''
This code fragments are divided into three parts:
- First, taking input and process the reference data samples clusterings
- Second, taking input of each random subsets of samples clusterings
- Finally, compare clusterings using the proposed metric

'''
#=============================================================================================
#=============================================================================================
# Taking input and preprocess of reference sample and reference sample clusterings
#=============================================================================================
#=============================================================================================
reference_dataFrame = pd.read_csv('reference_samples_80.txt', sep="\t", dtype=str, header=None)
...........................
...........................
# Considering only gene name column for mapping purpose
reference_gene_names = reference_gene_names[:, 0]
...........................
...........................


# Reference sample clusterings
dataFrame_all_samples_cls_res = pd.read_csv('exp_c9_reference.txt', sep="\t", header=None)
...........................
...........................

#=============================================================================================
#=============================================================================================
#The following loop is used here to map each reference sample gene onto its clusterings
#based on gene expression value.
#master_gene_class grouped together all genes based on its clusterings
#=============================================================================================
#=============================================================================================

master_gene_class = [[], [], [], [], [], [], [], [], []]

for i_genes in range(len(reference_gene_names)):
    if (int(reference_samples[i_genes]) == 1):
        master_gene_class[0].append(reference_gene_names[i_genes])

    if (int(reference_samples[i_genes]) == 2):
        master_gene_class[1].append(reference_gene_names[i_genes])

    if (int(reference_samples[i_genes]) == 3):
        master_gene_class[2].append(reference_gene_names[i_genes])
```

```

...........................
...........................

#=====================================================================================
#=====================================================================================
# Taking each random subsets of samples, preprocess, and compare clusterings
#    to reference sample clusterings
#=====================================================================================
#=====================================================================================

# For stemcell tissue data, total_subset_of_samples set to a range of 3 to 78.
# For Drosophila melanogaster data sample, the range of total_subset_of_samples will be 3 to 64.
for main_itr in range(total_subset_of_samples):
    sample_name = f"exp_c4_rand_{main_itr:d}.txt"


    # Preprocess each random subsets of samples
    dataFrame_cls_res = pd.read_csv(sample_name, sep="\t", header=None)
    total_iteration_time = len(dataFrame_cls_res)
    ...........................
    ...........................


    #================================================================================
    #================================================================================
    # Variable total_iteration_time is used here to maintain the iteration time.
    # Normally it set as 100. But for cluster size 8 to 10, we use single iteration time.
    #================================================================================
    #================================================================================
    for j_samples in range(total_iteration_time):

        gene_names = [[], [], [], [], [], [], [], [], []]


        # Map each subsets of samples gene to its clutering results.
        # Similar clustered genes are grouped into a single list.
        for i_genes in range(total_gene_size):
                if (int(random_samples_cluster_results[i_genes,j_samples]) == 1):
                    gene_names[0].append(reference_gene_names[i_genes])
                if (int(random_samples_cluster_results[i_genes, j_samples]) == 2):
                    gene_names[1].append(reference_gene_names[i_genes])
                if (int(random_samples_cluster_results[i_genes,j_samples]) == 3):
                    gene_names[2].append(reference_gene_names[i_genes])
                ...........................
                ...........................


        ...........................
        ...........................
        # Permuted clusterings are stored in all_permutation_result list.
        all_permutation_result = list(irt.permutations([gene_names[0], gene_names[1], gene_names[2], gene_names[3],...]))


        ...........................
        ...........................

        for pmut in range(len(all_permutation_result)):
            single_permuted_res = np.array(all_permutation_result[pmut])


            # Following loop is used here to find the best matches between two clusterings
            # and calculate the distance between two clusters.
            for com_m in range(len(all_permutation_result[pmut])):

                subset_results = single_permuted_res[com_m]
                ref_results = reference_cluster_result[com_m]

                # np.setdiff1d function returns unique elements between two lists.
                diff_number = np.setdiff1d(ref_results, subset_results)

                k = k + diff_number.size

            ...........................
            ...........................
            # Store the minimum distance in min_distance_diff
            if min_distance_diff > k:
                min_distance_diff = k

        sub_diff.append(min_distance_diff)

    main_diff.append(sum(i for i in sub_diff))


# main_diff contains the final distance score between a reference sample clustering
#   results compare to a random subsets of sample clusterings.
print(main_diff)


#==================================================
# Branch-and-bound
#==================================================

pmut = 0
        wh_range = len(all_permutation_result)

        while pmut < wh_range:

            single_permuted_res = np.array(all_permutation_result[pmut])
            if pmut == 0:
                k = 0
                for com_m in range(len(all_permutation_result[pmut])):
                    aa = single_permuted_res[com_m]
                    bb = reference_cluster_result[com_m]
                    diff_number = np.setdiff1d(bb, aa)
                    bu.append(diff_number.size)
                    k += diff_number.size
                min_distance_diff = k

            else:
                k = 0
                com_m = 0
                nes_wh = len(all_permutation_result[pmut])
                last_two_comb = nes_wh - 2
                while com_m < nes_wh:
                    aa = single_permuted_res[com_m]
                    bb = reference_cluster_result[com_m]
                    diff_number = np.setdiff1d(bb, aa)
                    bu.append(diff_number.size)
                    k += diff_number.size

                    if (k > min_distance_diff) and (com_m < last_two_comb):
                        index_var = len(all_permutation_result[pmut]) - com_m
                        new_var = int(math.factorial(index_var) / index_var)
                        pmut = pmut + (new_var - 1)
                        com_m = len(all_permutation_result[pmut])

                    com_m = com_m + 1

                if min_distance_diff > k:
                    min_distance_diff = k
            pmut += 1  # ==================================================
```

A.2 Bioinformatics tools

Reference genome file preprocessing and required software tools setup on Unix server.

```
# Download the reference Drosophila genome annotation file
$ url=”ftp://igenome:G3nom3s4u@ussd-ftp.illumina.com/Drosophila_melanogaster/Ensembl/BDGP5.25/Drosophila_melanogaster_Ensembl_BDGP5.25.tar.gz”
$ wget $url


$ mkdir GSE104073/Drosophila_melanogaster/
# Download the FASTQ file format from SRA
# For example, sample N1_heads_RNAseq sample SRA file is SRX3202288, and converted FASTQ file link is below:
$ ftp://ftp.sra.ebi.ac.uk/vol1/fastq/SRR605/009/SRR6055359/SRR6055359_1.fastq.gz
$ ftp://ftp.sra.ebi.ac.uk/vol1/fastq/SRR605/009/SRR6055359/SRR6055359_2.fastq.gz
...........................
...........................
# sample s24_WB_RNAseq
$ ftp://ftp.sra.ebi.ac.uk/vol1/fastq/SRR605/002/SRR6055422/SRR6055422_1.fastq.gz
$ ftp://ftp.sra.ebi.ac.uk/vol1/fastq/SRR605/002/SRR6055422/SRR6055422_2.fastq.gz


# Install required software tools:
$ sudo apt-get install bowtie
$ sudo apt-get install tophat
$ sudo apt-get install cufflinks
$ sudo apt-get install samptools


# Run TopHat protocol
$ tophat -p 8 –G genes.gtf -o 1_N1_heads_RNAseq_thout genome SRR6055359_1.fastq.gz SRR6055359_2.fastq.gz
...........................
...........................
$ tophat -p 8 –G genes.gtf -o 64_s24_WB_RNAseq_thout genome SRR6055422_1.fastq.gz SRR6055422_2.fastq.gz


# Run Cufflinks protocol
$ cufflinks -p 8 -o 1_N_heads_cout 1_N1_heads_RNAseq_thout/accepted_hits.bam
...........................
...........................
$ cufflinks -p 8 -o 64_S_WB_cout 64_s24_WB_RNAseq_thout/accepted_hits.bam
```

Appendix B

Clusterings comparison

Table B.1: Average clusterings comparison (K-means using Manhattan distance) over the 100 iterations, with standard error of the mean in parentheses for the mouse stem cell dataset.

Number of samples	Number of clusters 4	Number of clusters 5	Number of clusters 6	Number of clusters 7	Number of clusters 8	Number of clusters 9	Number of clusters 10
3	1433.9 (59.11)	1596.7 (107.15)	1475.6 (72.99)	2267.6 (84.17)	2645.6 (113.89)	3897.8 (113.22)	4448.2 (118.37)
4	833.5 (38.25)	1492.1 (98.83)	2070.3 (64.51)	1680 (87.28)	3279.1 (115.98)	3870.1 (114.24)	3957.6 (121.43)
5	1061.3 (21.52)	1320.6 (91.05)	1757.1 (59.88)	1391.9 (69.89)	2697.2 (105.36)	4184.2 (132.43)	3901 (114.96)
6	796 (52.18)	1832.3 (94.13)	1098.5 (66.46)	2222.1 (84.98)	3148.9 (124.22)	2921.8 (133.61)	3577.6 (122.96)
7	784.4 (42.71)	1812.2 (114.51)	1486 (67.43)	1584.6 (77.25)	3296 (120.29)	3533.8 (140.95)	3315.9 (134.27)
8	839.7 (80.29)	1943.3 (113.79)	1315.8 (66.65)	1788.7 (71.35)	3079 (110.63)	3138.1 (138.13)	4469.8 (135.31)
9	566 (54.73)	2340.4 (104.49)	1505.4 (68.47)	1394.6 (78.54)	3184.8 (124.93)	4015.2 (142.42)	3094.8 (135.18)
10	509.1 (30.52)	1525.4 (113.12)	1349.7 (67.52)	1776.6 (83.41)	2676 (111.17)	2566.6 (148.6)	4018.6 (135.37)
11	544 (49.96)	1387.7 (108.47)	1706.1 (68.99)	1168.6 (83.14)	1989.2 (115.23)	3471.8 (154.65)	3898.4 (141.51)
12	545.8 (51.12)	1345.3 (104.62)	1390.6 (69.05)	1603 (79.5)	3002.8 (115.93)	3813.1 (155.51)	3650.5 (136.08)
13	845.1 (13.95)	1507 (111.6)	1212.9 (61.96)	1154.2 (92.47)	2937.8 (125.11)	3121.9 (158.48)	3335 (133.44)
14	481.4 (33.98)	1378.2 (110.88)	1080.2 (63.17)	1820.9 (80.09)	3404.4 (127.84)	4033.1 (161.33)	3411.8 (115.18)
15	523.3 (16.76)	1282.8 (109.11)	1238.1 (66.75)	1677.3 (86.14)	2900.6 (123.86)	3382.4 (150.65)	3359.3 (139.24)
16	675.3 (60.76)	1209 (122.66)	1705.6 (67.16)	1533.1 (85.24)	2905.8 (133.1)	3448.2 (153.89)	2658.6 (140.63)
17	382.5 (28.52)	1465.5 (112.46)	1377.6 (71.47)	1306.5 (80.73)	2698.5 (129.96)	2652.2 (158.73)	2985.2 (131.84)
18	440.7 (32.68)	1087.3 (117.48)	1197.8 (72.61)	1551.8 (74.03)	3156.7 (134.58)	2691 (153.47)	3223.7 (143.87)
19	446.2 (47.17)	1171.1 (105.06)	1430.8 (71.22)	1105.7 (84.66)	2620.4 (129.07)	4259.3 (167.76)	3175 (147.37)
20	471.7 (14.78)	1361.7 (112.16)	1327.5 (69.34)	1312.3 (87.28)	2093.6 (127.58)	3437.7 (157.78)	3514.5 (152.63)
21	316.2 (45.35)	1906 (105.55)	1498.2 (65.37)	1282.5 (84.9)	2938.9 (128.62)	2768 (159.99)	3756.8 (134.33)
22	808.1 (14.24)	1431.9 (112.27)	1230.9 (70.55)	1430.6 (94.06)	3375.7 (133.82)	2977.4 (162.13)	3510.6 (143.84)

Table B.1 continued from previous page

Number of samples	Number of clusters 4	Number of clusters 5	Number of clusters 6	Number of clusters 7	Number of clusters 8	Number of clusters 9	Number of clusters 10
23	420.8 (12.05)	1265.4 (111.39)	966.6 (68.11)	671 (80.89)	2609 (126.58)	3228.6 (174.77)	4262.3 (140.76)
24	367.2 (32.79)	1041.1 (118.81)	1364.8 (71.98)	1038.9 (80.22)	2782.4 (129.28)	3582.5 (161.85)	3641.8 (150.43)
25	297.9 (11.05)	1231.6 (112.86)	1180.3 (67.94)	1364.3 (90.21)	2725.6 (131.88)	3975.9 (165.22)	3590.5 (140.69)
26	339.1 (10.83)	1146.5 (127.86)	1103.2 (75.95)	1161.6 (84.62)	2842.2 (128.78)	3212 (172.26)	3467.4 (149.04)
27	351.4 (16.21)	1380.3 (111.75)	1237.1 (73.08)	1347.6 (86.85)	2320.7 (132.11)	3351.9 (166.11)	3628 (155.9)
28	396.4 (11.13)	1736.6 (116.22)	1269.3 (74.51)	1122.4 (85.94)	2222.8 (134.81)	2910.5 (166.75)	3855.7 (161.11)
29	329.2 (10.77)	1611.1 (121.06)	1495.9 (74.78)	1415.3 (82.37)	2364.6 (138.53)	2251.5 (178.68)	3117.9 (162.37)
30	271.4 (12.62)	625.3 (123.5)	1572.5 (72.39)	1586.3 (82.99)	2751.4 (138.72)	2550.3 (168.86)	3694.4 (154.28)
31	375.8 (10.6)	1576.9 (121.55)	1285.9 (78.08)	1357.1 (90.05)	2216.1 (137.97)	2450.2 (163.32)	3257.4 (164.08)
32	285 (11.6)	1675.2 (123.21)	956.1 (78.55)	1287.7 (82.86)	2965 (137.85)	3083.2 (171.51)	2992.3 (156.69)
33	283.5 (11.1)	1182.9 (110.23)	1442.7 (75.69)	1428 (92.23)	2125.7 (123.82)	2217.3 (168.56)	4355.5 (152.77)
34	345.4 (11.2)	1711.3 (113.21)	810.1 (79.77)	1330.6 (78.66)	2652.1 (124.75)	3464.3 (169.6)	3543 (144.31)
35	306.6 (12.69)	1109.5 (125.42)	1041.7 (77.15)	1076.3 (87.96)	2120.1 (139.85)	3154.7 (170.5)	3249.4 (144.31)
36	267.1 (12.58)	1438.5 (107.79)	1092.8 (77.61)	1021.2 (77)	2960.6 (122.76)	3044.7 (171.73)	3766 (162.24)
37	331.1 (10.78)	1191.8 (105.68)	1277 (82.29)	908.8 (90.9)	2358.7 (135.76)	4173.1 (171.96)	4482.7 (160.1)
38	292.6 (12.4)	521.7 (126.89)	1200.5 (74.82)	1052.6 (82.46)	3028.7 (137.88)	2521.4 (178.43)	4413.1 (159.93)
39	226.9 (10.71)	1523 (115.37)	1097.8 (79.15)	804.3 (88.91)	2520.6 (124.35)	3827.5 (179.62)	3396.3 (166.12)
40	232.1 (11.69)	986.8 (121.63)	883.7 (77.97)	1146.3 (87.73)	2190.5 (126.77)	3844.8 (168.86)	3485.1 (136.93)
41	279.3 (11.77)	1493.6 (114.22)	1316.4 (71.88)	1341.1 (79.29)	3187.5 (134.32)	3453.2 (175.69)	3678.6 (124.12)
42	334.3 (11.35)	1406.8 (120.56)	1193 (80.2)	833.3 (87)	2828.1 (135.65)	3548.3 (180.23)	3157.3 (149.32)
43	184.4 (12.6)	1714.7 (119.64)	1754.5 (76.52)	1444.7 (92.28)	1731.7 (131.01)	2600.7 (182.5)	3285.1 (144.49)
44	235.9 (10.75)	1742.4 (123.51)	989.2 (78.35)	841.7 (86.4)	2226.8 (128.83)	2608.2 (181.16)	3794.3 (171.24)
45	299.5 (12.2)	1075.9 (120.11)	1345.9 (83.67)	1188.4 (84.15)	2666.5 (121.91)	4185.2 (184.62)	3059.1 (156.1)
46	209.8 (13.09)	1779.7 (126.75)	1159.5 (82.23)	1065.2 (85.68)	2486.9 (125.77)	3397 (173.59)	3252.5 (140.55)
47	283.7 (11.47)	1901.4 (115.41)	780.5 (78.38)	1136.8 (87.79)	3057.6 (140.28)	3827.1 (170.84)	3232.2 (160.55)

Table B.1 continued from previous page

Number of samples	Number of clusters 4	Number of clusters 5	Number of clusters 6	Number of clusters 7	Number of clusters 8	Number of clusters 9	Number of clusters 10
48	209.2 (13.05)	1869.9 (120.48)	630.5 (80.17)	1309 (87.29)	2113.6 (136.38)	3424.5 (159.95)	3150.8 (153.69)
49	230 (14.04)	1392.8 (113.69)	954.4 (82.86)	1176.9 (87.03)	2276.1 (138.65)	3713.4 (177.42)	3856.2 (162.32)
50	260.7 (10.57)	266.7 (124.42)	875.8 (79.72)	777.8 (89.38)	2159.9 (130.64)	1954.5 (170.42)	3059 (164.8)
51	274.5 (10.13)	1044.4 (115.38)	1499.9 (81.1)	1169.2 (87.39)	3220.5 (130.7)	2639.9 (173.37)	2772 (159.71)
52	227 (13.04)	1658.7 (122.04)	1080.6 (84.98)	1105.9 (76.66)	2662.7 (133.43)	4088.6 (175.02)	3089.4 (171.9)
53	217.2 (12.04)	826.1 (119.43)	1016.5 (78.69)	673.2 (85.36)	2783.3 (121.91)	2592.9 (182.84)	3857.4 (167.58)
54	290.7 (12.53)	1014.8 (123.41)	1177.8 (76.01)	1215.7 (85.03)	1987.4 (140.91)	2116.8 (177.81)	3286.2 (157.25)
55	127.9 (11.17)	1909.1 (120.28)	1375.8 (79.7)	1216.9 (86.3)	2651.5 (135.45)	3645.4 (172.09)	3446.8 (165.76)
56	325 (13.21)	616.1 (124.92)	1344.4 (78.68)	1442.8 (87.77)	2863.4 (132.41)	3190.4 (170.37)	3779 (155.49)
57	232.2 (12.23)	882 (125.2)	628.9 (85.49)	1308.4 (83.6)	2799.6 (135.01)	3350.4 (185.92)	4045.5 (159.27)
58	212.2 (12.49)	1427.6 (125.73)	562.9 (82.8)	1387.8 (91.42)	2147.7 (121.96)	2489.7 (186.95)	4046.2 (154.55)
59	140.2 (11.75)	1678.7 (124.63)	848.4 (82.32)	1549.1 (90.42)	2269.7 (138.57)	2714.4 (182.22)	2847.7 (158.98)
60	234.5 (11.69)	1509.1 (124.16)	1133.1 (83.31)	1044.8 (84.59)	2130.4 (136.81)	3257.8 (182.58)	3525.5 (158.13)
61	170 (12.52)	1121.9 (123.35)	1482.5 (82.75)	995.8 (84.18)	2352.5 (131.71)	3682.8 (181.71)	2538.2 (154.73)
62	117.7 (12.71)	774.8 (125.56)	1080.9 (83.68)	1141.7 (83.26)	3480.6 (135.31)	3494.3 (186.07)	2877.1 (139.54)
63	157.9 (12.21)	1551.2 (127.62)	866.9 (81.79)	636.8 (85.65)	3003.7 (136.03)	3205.5 (177.95)	4410.6 (130.56)
64	153.8 (13.67)	1144.1 (121.91)	1402.8 (83.23)	325.4 (82.47)	3338 (138.18)	3146.5 (183.1)	3159.1 (170.58)
65	159.9 (11.11)	1320.3 (130.24)	1129.4 (84.31)	761.1 (87.65)	2374 (136.92)	3435 (183.29)	3641.4 (171.16)
66	135.4 (11.56)	958.8 (120.2)	1475.8 (82.62)	1248 (84.54)	3350 (139.2)	2607.6 (181.15)	4267.7 (163.32)
67	111.5 (11.05)	1493.2 (124.63)	845.4 (81.66)	833.6 (86.78)	2636.1 (125.28)	2924.9 (176.3)	4330.5 (155.07)
68	99.4 (12.04)	692.7 (127.92)	1296.2 (86.58)	1393.7 (88.81)	2160.9 (138.5)	2306.8 (184.83)	4245 (159.75)
69	101.8 (13.16)	2133.6 (123.65)	1007.4 (84.11)	1268.1 (85.43)	3142 (139.17)	3250.1 (181.69)	3411.7 (146.41)
70	84.1 (14.02)	995.4 (127.67)	1275.5 (88.68)	1604 (80.08)	3094.1 (124.21)	2879.3 (176.14)	4320.7 (150.97)
71	156.6 (13.2)	1232 (119.1)	1184.7 (89.39)	1161.8 (84.49)	2967 (141.05)	2435.9 (180.58)	2469.9 (147.54)
72	106.1 (10.31)	836.7 (127.08)	1468.8 (85.59)	1224.2 (84.83)	2861.6 (136.28)	3457.4 (182.16)	3804.9 (150.57)

Table B.1 continued from previous page

Number of samples	Number of clusters 4	Number of clusters 5	Number of clusters 6	Number of clusters 7	Number of clusters 8	Number of clusters 9	Number of clusters 10
73	100.8 (12.43)	764.3 (125.18)	1066.5 (87.86)	783.8 (82.03)	2346.1 (133.98)	3699.9 (193.78)	3343.4 (150.81)
74	93.1 (12.63)	1891.8 (122.48)	967.5 (88.44)	1154.3 (84.35)	2569 (141.76)	3358.1 (198.49)	3864.9 (150.73)
75	133.5 (9.45)	1117.5 (129.07)	1134.5 (91.08)	729.9 (84.12)	2856 (131.55)	2601.9 (167.72)	3816.3 (166.97)
76	49.5 (10.66)	1213.7 (125.7)	1215.4 (87.82)	895 (84.17)	2710.1 (140.12)	3001.6 (187.32)	2644 (157.88)
77	69.8 (8.78)	1782.8 (127.63)	1331.1 (84.41)	771.4 (82.87)	3391.6 (138.71)	3305 (189.97)	3538.5 (170.21)
78	80 (10.12)	1730.4 (125.36)	981.5 (87.68)	1811.5 (84.44)	2686.1 (122.88)	2677.4 (184.92)	3521.8 (150.61)

Table B.2: Average clusterings comparison (K-means using Manhattan distance) over the 100 iterations, with standard error of the mean in parentheses for the *Drosophila melanogaster* dataset.

Number of samples	Number of clusters 4	Number of clusters 5	Number of clusters 6	Number of clusters 7	Number of clusters 8	Number of clusters 9	Number of clusters 10
3	1287.59 (64.83)	1651.1 (89.52)	1927.85 (97.55)	2999.33 (96.38)	3422.97 (82.63)	3981.69 (100.32)	4006.13 (100.23)
4	1217.24 (56.85)	1550.9 (73.06)	1685.06 (97.62)	2528.75 (88.84)	2788.47 (93.92)	3460.4 (103.16)	3570.95 (104.32)
5	1088.23 (53.74)	1305.15 (56.49)	1521.82 (86.16)	2242.66 (87.85)	2563.76 (100.12)	3221.26 (98.7)	3413.03 (93.44)
6	1065.29 (45.83)	1260.72 (68.08)	1297.65 (81.62)	2269.21 (83.26)	2297.95 (94.14)	3021.47 (86.35)	2971.53 (91.33)
7	1018.5 (46.25)	1203.19 (59.65)	1450.21 (74.74)	2044.12 (87.76)	2292.19 (95.23)	2959.5 (89.18)	2835.62 (84.57)
8	984.18 (49.18)	1056.43 (62.98)	1292.67 (72.18)	2059.63 (63.04)	1963.65 (83.53)	2717.21 (87.11)	2702.01 (90.03)
9	1008.13 (43.04)	1110.58 (50.9)	1162.91 (65.37)	1980.56 (64.98)	1965.45 (71.05)	2669.39 (76.72)	2393.89 (71.43)
10	947.43 (34.87)	1079.9 (37.53)	1080.05 (51.33)	1893.51 (67.21)	1799.63 (66.69)	2563.99 (69.29)	2532.86 (92.63)
11	1032.19 (34.68)	1166.82 (46.12)	1066.01 (50.2)	1818.63 (54.12)	1824.66 (59.9)	2476.56 (60.28)	2332.25 (78.53)
12	865.47 (57.78)	1015.23 (48.7)	1047.34 (55.95)	1826.33 (55.03)	1900.38 (65.23)	2381.46 (71.78)	2311.02 (72.65)
13	912.15 (41)	1021.87 (41.54)	1060.28 (57.38)	1747.14 (49.96)	1911.54 (69.69)	2537.67 (74.93)	2191.46 (64.1)
14	953.36 (36.32)	987.03 (44.57)	1068.33 (52.27)	1797.86 (62.21)	1871.53 (75.81)	2297.36 (61.98)	2264.53 (65.7)
15	839.65 (29.6)	1095.04 (53.11)	925.85 (46.51)	1733.32 (55.6)	1675.82 (55.54)	2238.96 (59.47)	2337.14 (78.66)
16	922.59 (30.51)	993.52 (44.63)	1038.56 (65.7)	1804.17 (65.53)	1716.89 (64.81)	2313.38 (56.04)	2224.98 (67.19)
17	942.32 (63.96)	912.56 (40.2)	963.56 (58.95)	1802.96 (54.78)	1724.67 (52.8)	2233.83 (60.91)	2272.28 (73.66)
18	852.58 (40.98)	1005.45 (31.43)	915.14 (53.3)	1691.71 (57.75)	1722.24 (55.66)	2189.47 (49.68)	2130.33 (62.92)

Table B.2 continued from previous page

Number of samples	Number of clusters 4	Number of clusters 5	Number of clusters 6	Number of clusters 7	Number of clusters 8	Number of clusters 9	Number of clusters 10
19	971.78 (40.93)	909.4 (61.23)	961.1 (60.85)	1725.16 (55.9)	1653.14 (61.62)	2280.35 (63.53)	2114.68 (66.07)
20	859.11 (37.98)	946.83 (38.35)	888.36 (48.75)	1689.05 (53.58)	1727.57 (53.89)	2191.11 (61.65)	2099.69 (52.4)
21	862.74 (31.24)	897.77 (43.44)	894.24 (64.03)	1647.91 (46.72)	1591.43 (52.6)	2147.65 (55.78)	2133.03 (68.95)
22	866.91 (30.38)	854.24 (39.57)	844.47 (41.7)	1702.35 (51.63)	1675.77 (62.63)	2239.74 (59.37)	2039.99 (57.88)
23	913.9 (27.4)	980.85 (58.61)	1001.22 (47.33)	1632.22 (43.04)	1599.62 (57.39)	2185.75 (57.75)	1895.86 (57.12)
24	885.82 (61.62)	933 (45.21)	871.23 (47.47)	1604.42 (48.82)	1633.33 (72.03)	2188.8 (54.02)	1940.11 (57.64)
25	811.41 (58.62)	894.78 (53.58)	880.53 (86.82)	1674.31 (63.54)	1538.98 (49.48)	2139.9 (63.19)	1939.44 (50.75)
26	907.45 (30.4)	942.15 (40.24)	900.02 (51.12)	1676.9 (52.42)	1607.14 (65.4)	2185.41 (53.55)	1962.3 (61.11)
27	858.64 (71.83)	963.72 (49.5)	992.37 (41.09)	1623.12 (59.32)	1742.66 (88.06)	2280.96 (87.27)	2035.86 (72.92)
28	969.58 (81.08)	1043.56 (60.89)	909.95 (41.25)	1763.4 (77.55)	1621.21 (61.86)	2033.57 (45.18)	1912.12 (53.54)
29	835.7 (37.19)	1051.84 (68.51)	951.49 (40.25)	1650.44 (76.39)	1562.75 (46.38)	2161.29 (68.61)	1901.41 (67.45)
30	837.78 (69.38)	945.57 (54.48)	796.8 (36.35)	1662.29 (63.51)	1573.69 (68)	2050.57 (38.66)	1950.98 (59.56)
31	780.76 (70.24)	929.53 (50.57)	1060.88 (59.47)	1614.73 (62.78)	1491.81 (37.3)	2189.34 (62.17)	1860.2 (66.54)
32	857.09 (60.61)	942.27 (48.66)	926.87 (54.78)	1602.06 (53.51)	1581.21 (69.35)	1982.54 (43.77)	1795.18 (46.58)
33	861.49 (27.04)	978.84 (73.23)	905.66 (48.75)	1644.98 (59.67)	1668.24 (81.55)	2102.91 (66.02)	2012.96 (93.47)
34	975.03 (53.58)	1031.43 (38.41)	862.54 (49.01)	1547.05 (72.75)	1479.95 (58.28)	2124.77 (52.8)	1772.8 (55.72)
35	866.18 (70.48)	903.3 (48.09)	829.53 (52.59)	1642.42 (55.38)	1477.28 (80.82)	2182.88 (81.21)	1856.81 (44.52)
36	848.88 (74.43)	1008.91 (75.31)	954.04 (63.35)	1598.82 (70.48)	1507.43 (60.12)	1992.98 (40.92)	1920.03 (85.23)
37	937.6 (29.8)	884.7 (51.06)	1036.77 (57.96)	1661.5 (76.83)	1584.29 (88.61)	1989.29 (57.61)	1796.47 (60.99)
38	825.02 (81.31)	942.1 (78.64)	945.24 (100.67)	1526.77 (69.2)	1493.91 (63.87)	2076.42 (60.85)	1969.22 (82.02)
39	930.68 (95.57)	902.32 (77.73)	854.18 (73.75)	1786.31 (85)	1634.26 (75.48)	2124.06 (65.18)	1804.88 (62)
40	974.51 (62.69)	854.55 (100.84)	1024.5 (76.3)	1686.5 (72.05)	1530.1 (63.48)	2122.15 (81.38)	1794.15 (50.19)
41	859.73 (59.93)	1027.99 (79.83)	974.12 (77.63)	1634.66 (73.74)	1440.68 (51.91)	2133.38 (66.77)	1809.88 (56.62)
42	1010.16 (54.93)	849.33 (66.65)	904.78 (93.37)	1557.36 (74.14)	1537.27 (59.91)	2127.57 (73.8)	1947.67 (98.11)
43	898.24 (66.13)	944.58 (86.49)	895.99 (65.55)	1550.59 (77.04)	1630.63 (97.3)	2117.48 (92.29)	1808.15 (85.46)

Table B.2 continued from previous page

Number of samples	Number of clusters 4	Number of clusters 5	Number of clusters 6	Number of clusters 7	Number of clusters 8	Number of clusters 9	Number of clusters 10
44	906.54 (77.86)	1103.12 (45.54)	773.43 (65.94)	1612.61 (81.29)	1427.18 (61.79)	2128.19 (91.37)	2017.47 (97.12)
45	1014.25 (61.01)	1002.18 (126.67)	995.11 (75.92)	1739.99 (81.59)	1468.25 (77.31)	2146.82 (92.87)	1832.82 (71.17)
46	785.47 (96.54)	1018.25 (100.89)	1000.92 (46.15)	1869.5 (65.42)	1470.93 (57.63)	2182.15 (98.53)	1880.17 (81.97)
47	1162.85 (62.3)	950.33 (98.15)	1005.08 (112.7)	1639.77 (88.66)	1613.54 (88)	2159.14 (96.09)	1916.36 (94.75)
48	908.13 (117.89)	1127.22 (63.37)	939.76 (88.18)	1619.76 (106.14)	1782.91 (103.77)	2131.88 (87.93)	1984.96 (98.49)
49	945.83 (90.51)	998.44 (98.26)	835.32 (123.41)	1591.6 (106.46)	1747.5 (113.19)	2101.1 (88.28)	1962.72 (93.29)
50	893.41 (62.61)	920.82 (115.88)	833.75 (133.96)	1821.95 (95.27)	1830.87 (127.04)	2167.44 (88.48)	1950.74 (89.94)
51	1058.83 (129.39)	921.4 (110.27)	1103.16 (114.95)	1857.06 (99.25)	1705.25 (121.94)	2099.65 (83.74)	1861.94 (80.58)
52	929.2 (105.4)	1173.05 (114.06)	1054.94 (109.54)	1732.49 (104.53)	1592.36 (83.63)	2199.9 (98.37)	1920.13 (106.97)
53	1004.42 (49.82)	1008.24 (92.45)	987.78 (107.32)	1736.95 (130.63)	1572.71 (91.16)	2321.41 (111.28)	1889.97 (100.52)
54	1224.26 (81.77)	990.56 (100.4)	988.1 (136.83)	1671.81 (125.02)	1672.36 (115.23)	2241.63 (112.2)	2114.5 (115.9)
55	979.16 (110.84)	1001.71 (116.53)	1098.29 (102.82)	1834.62 (111.24)	1695.54 (118.97)	2210.8 (107.64)	1864.52 (120.87)
56	993.57 (88.26)	984.5 (96.82)	946.12 (115.64)	1646.45 (111.72)	1581.72 (104)	2473.51 (141.92)	2080.51 (124.11)
57	851.17 (107.83)	1208.41 (99.36)	1042.06 (117.95)	1903.33 (122.52)	1688.51 (110.89)	2204.25 (108.17)	2076.27 (127.41)
58	1251.27 (54.63)	950.28 (105.35)	1168.83 (122.26)	1858.58 (106.01)	1636.34 (97.23)	2206.72 (104.83)	2023.01 (117.7)
59	960.43 (83.52)	1009.96 (108.97)	1106.87 (120.91)	1757.45 (112.17)	1640.34 (103.1)	2375.37 (117.32)	2009.3 (119.14)
60	978.28 (127.37)	912.34 (142.49)	1340.51 (104.16)	1646.6 (101.37)	1612.66 (107.03)	2340.11 (120.67)	2065.33 (127.87)
61	1010.01 (90.23)	1215.89 (110.45)	937.95 (129.13)	1824.85 (127.47)	1658.64 (128.41)	2178.82 (104.12)	1973.96 (111.89)
62	1050.57 (69.04)	1052.47 (133.36)	943.1 (106.48)	2009.19 (133.46)	1912.59 (142.31)	2549.17 (150.84)	2316.69 (145.53)
63	860.03 (152.23)	894.14 (111.71)	1205.99 (137.37)	1585.1 (140.82)	1791.8 (131.03)	2263.91 (123.94)	2050.69 (130.95)
64	1244.07 (132.36)	1290.53 (155.38)	1153.73 (107.15)	1834 (133.33)	1741.5 (133.52)	2332.36 (126.43)	2114.64 (132.17)

Table B.3: Average clusterings comparison (K-means using Euclidean distance) over the 100 iterations, with standard error of the mean in parentheses for the mouse multi-tissue dataset.

Number of samples	Number of clusters 4	Number of clusters 5	Number of clusters 6	Number of clusters 7	Number of clusters 8	Number of clusters 9	Number of clusters 10
3	544.4 (10.58)	988.9 (14.43)	3990.4 (33.72)	5474.7 (49.86)	729.85 (67.19)	989.53 (95.8)	1218 (117.05)

Table B.3 continued from previous page

Number of samples	Number of clusters 4	Number of clusters 5	Number of clusters 6	Number of clusters 7	Number of clusters 8	Number of clusters 9	Number of clusters 10
4	445.1 (5.91)	950.7 (17.33)	3072.7 (15.97)	5140.2 (40.26)	634.25 (59.12)	666.92 (60.77)	833.04 (73.69)
5	309.1 (5.72)	1024.9 (18.38)	3040.7 (22.42)	4280.2 (32.86)	420.87 (41.68)	534.82 (42.3)	940.33 (85.92)
6	302.5 (4.9)	842.4 (18.45)	2845.7 (16.12)	3630.3 (20.26)	471.66 (44.82)	505.56 (46.97)	686.9 (62.36)
7	252.2 (2.84)	543.2 (9.66)	2709 (7.02)	3491.6 (14.74)	329.16 (30.67)	503.04 (40.95)	662.01 (60.21)
8	231.3 (2.7)	517.5 (6.92)	2766.6 (8.26)	3449.8 (15.33)	326.79 (34.47)	517.77 (48.43)	508.91 (44.19)
9	236.5 (1.7)	454.1 (5.8)	2746.2 (10.94)	3451.2 (12.64)	295.65 (32.32)	392.93 (28.23)	514.12 (41.75)
10	215.4 (2.43)	439.5 (5.44)	2677.8 (5.66)	3300 (10.23)	271.39 (26.59)	401.82 (30.26)	524.89 (47.58)
11	181.1 (1.49)	600.5 (7.93)	2695.4 (5.63)	3519.8 (12.54)	310.81 (36.14)	354.2 (20.73)	462.24 (38.54)
12	170.5 (1.31)	489.5 (6.53)	2700 (5.37)	3321 (11.81)	273.04 (25.55)	332.7 (23.39)	510.47 (45.41)
13	172.4 (1.41)	411.4 (5.61)	2666.9 (7.96)	3071.5 (10.97)	250.57 (18.9)	358.38 (20.48)	423.15 (34.18)
14	211.3 (1.77)	432 (5.85)	2574.1 (5.72)	3005.1 (11.43)	233.49 (25.01)	346.14 (23.24)	355.6 (25.49)
15	204 (1.58)	359 (3.42)	2618.7 (6.03)	3376.2 (13.54)	243.13 (29.2)	342.82 (22.2)	426.88 (35.5)
16	194.1 (1.4)	480 (6.76)	2686.5 (5.93)	3463 (9.97)	196.47 (20.27)	343.57 (20.51)	506.88 (39.62)
17	178.1 (1.45)	286.2 (2.89)	2592 (6.93)	3252 (11.03)	249.54 (21.08)	317.86 (20.44)	354 (25.49)
18	167.2 (1.21)	350.1 (4.22)	2485.7 (7.71)	3281.3 (11.07)	232.8 (26.79)	303.74 (14.88)	396.01 (30.04)
19	269.2 (4.14)	293 (3.32)	2659.8 (5.89)	3214.4 (11.23)	252.87 (22.09)	290.95 (16.72)	402.71 (33.25)
20	148.6 (1.29)	286.5 (3.48)	2517 (6.89)	3320.1 (10.66)	236.42 (21.12)	312.88 (20.47)	336.51 (24.77)
21	153.9 (1.52)	379.3 (5.83)	2698.8 (5.35)	3553.3 (9.38)	238.75 (24.39)	282.63 (18.32)	344.69 (20.62)
22	188.1 (1.52)	308.7 (4.29)	2568.2 (6.97)	3513.6 (10.58)	214.14 (18.55)	286.81 (16.11)	334.12 (30.2)
23	197.9 (1.76)	409 (6.14)	2596.4 (7.02)	3285.8 (10.64)	223.85 (20.44)	307.35 (16.9)	331.25 (22.68)
24	185.9 (1.47)	381.7 (5.76)	2577.8 (7.04)	3418 (10.34)	226.11 (27.12)	289.66 (16.81)	377.5 (28.75)
25	177.1 (1.54)	350.1 (5.55)	2556.1 (7.59)	3422.1 (10.54)	225.06 (20)	302.63 (16.61)	330.56 (21.41)
26	183.2 (1.5)	500 (8.99)	2520.1 (7.24)	3140.5 (12.19)	190.29 (18.15)	246.31 (12.56)	338.19 (26.11)
27	150.1 (1.46)	387.9 (6.89)	2602.2 (6.89)	3254.1 (11.79)	198.63 (18.51)	269.1 (16.07)	401 (35.9)
28	166.3 (1.28)	315.8 (3.67)	2637.9 (6.43)	3181.1 (11.77)	194.32 (20.9)	285 (16.77)	317 (24.18)

Table B.3 continued from previous page

Number of samples	Number of clusters 4	Number of clusters 5	Number of clusters 6	Number of clusters 7	Number of clusters 8	Number of clusters 9	Number of clusters 10
29	164.8 (1.41)	289.8 (3.16)	2756.7 (5.49)	3385.8 (11.18)	249.87 (22.47)	258.39 (12.59)	286.1 (19)
30	158.7 (1.47)	288.6 (2.16)	2679.1 (6.6)	3452.8 (10.27)	183.07 (17.44)	277.39 (18.15)	308 (17.62)
31	172.9 (1.6)	289.8 (2.66)	2607.9 (7.05)	3107 (13.3)	224.31 (18.97)	267.61 (20.26)	339 (27.87)
32	231.7 (4.7)	415 (6.67)	2777.9 (4.89)	3359.4 (11.66)	194.16 (14.85)	275.32 (17.92)	292.18 (16.75)
33	176.2 (1.99)	320.2 (3.85)	2613 (6.95)	3561.5 (9.99)	175.96 (15.17)	264.77 (14.84)	335.02 (27.87)
34	161.2 (1.38)	367.7 (6.19)	2772.4 (5.48)	3390.8 (11.81)	197.8 (17.02)	266.32 (17.39)	248.81 (18.1)
35	177.2 (1.54)	467.7 (7.35)	2680.1 (6.5)	3214.3 (12.58)	179.8 (17.24)	253.98 (14.36)	309.83 (24.56)
36	160.7 (1.48)	293.7 (3.51)	2761.5 (7.21)	3009.9 (14.26)	160.97 (14.13)	245.56 (14.15)	299.09 (19.88)
37	147.8 (1.23)	307 (2.78)	2756.1 (5.82)	3157.4 (13.4)	184.99 (14.7)	289.7 (14.9)	295.83 (17.98)
38	182.9 (1.58)	275.9 (3.7)	2685 (6.85)	3319.7 (12.08)	174.64 (16.06)	281.45 (20.92)	315.79 (26.34)
39	157.7 (1.26)	333.6 (4.65)	2705.1 (6.2)	3375.4 (12.32)	182.05 (16.62)	247.26 (16.09)	316 (24.02)
40	168.5 (1.44)	390.9 (5.88)	2628.2 (7.54)	3274.5 (11.97)	158.28 (14.02)	257.22 (22.57)	304.79 (23.87)
41	143.7 (1.42)	443.7 (7.12)	2621.6 (6.96)	3344.5 (12.3)	196.23 (19.25)	246.27 (15.21)	297.4 (23.69)
42	162.1 (1.4)	284.6 (3.95)	2802.6 (4.17)	3296 (12.42)	226.32 (23.58)	226.96 (18.4)	286.5 (21.14)
43	135.7 (1.23)	250.7 (1.87)	2692.9 (6.16)	3479.1 (11.66)	183.58 (16.8)	284.47 (17.19)	266.75 (17.51)
44	160.6 (1.38)	261.8 (1.92)	2659.1 (7.23)	3242 (12.56)	203.35 (17.39)	250.12 (18.48)	260.67 (16.46)
45	177.9 (1.57)	351.9 (4.77)	2717.6 (6.09)	3217.1 (12.93)	202.87 (16.94)	250.86 (17.31)	267.3 (14.37)
46	162.7 (1.46)	391.2 (6.54)	2746.9 (5.94)	3089 (13.97)	236.24 (19.97)	296.67 (15.7)	248.57 (12.38)
47	168.2 (1.5)	335.8 (4.93)	2701.1 (6.25)	3315.1 (13.2)	186.09 (20.44)	250.48 (15.5)	299.65 (21.26)
48	138.5 (1.31)	272.1 (3.2)	2675.8 (6.53)	3406.1 (12.35)	204.9 (16.79)	267.82 (15.65)	272.83 (19.5)
49	162.1 (1.35)	334 (5.91)	2628.9 (7.53)	3423.8 (12.31)	193.54 (14.96)	257.6 (15.42)	285 (23.16)
50	148.9 (1.33)	241.6 (1.87)	2668.3 (7.05)	3279.5 (13.65)	198.58 (20.93)	256.41 (14.36)	304.84 (24.29)
51	144.4 (1.47)	418.8 (6.26)	2627.3 (7.29)	3126.3 (14.52)	212.87 (15.97)	244.02 (16.25)	234 (15)
52	157.3 (1.38)	426.3 (6.88)	2737.6 (6.12)	3119.8 (13.44)	180.61 (17.19)	260.97 (15.65)	265.78 (17.48)
53	178.6 (3.18)	353.6 (4.98)	2617 (7.05)	3392.9 (16.01)	184.66 (19.4)	264.05 (17.72)	290 (23.71)

Table B.3 continued from previous page

Number of samples	Number of clusters 4	Number of clusters 5	Number of clusters 6	Number of clusters 7	Number of clusters 8	Number of clusters 9	Number of clusters 10
54	171 (1.66)	290.5 (3.72)	2704.6 (6.29)	3264.3 (13.75)	167.02 (18.39)	222.47 (15.88)	287.96 (23.99)
55	153.6 (1.52)	359.9 (5.31)	2757.8 (6.01)	3350 (12.65)	192.68 (15.18)	229.12 (14.91)	256.04 (20.74)
56	159.5 (1.53)	348.7 (5.16)	2743.3 (5.91)	3057.3 (14.61)	161.15 (14.57)	244.74 (17.52)	302.3 (22.3)
57	164.6 (1.39)	346.1 (5.93)	2747.7 (6.23)	3278.9 (13.59)	171.99 (17.5)	246.5 (16.4)	268.67 (18.45)
58	154 (1.49)	239.9 (3.03)	2723.1 (6.36)	3319.6 (12.62)	191.9 (18.18)	262.92 (15.36)	236.14 (13.85)
59	141.8 (1.4)	352.7 (6.13)	2696.2 (5.91)	3359.3 (13.11)	223.97 (23.37)	247.48 (16.26)	244.11 (16.24)
60	140.7 (1.37)	390.1 (6.51)	2679.6 (7.26)	3479.4 (12.88)	180.51 (17.18)	257.79 (15.57)	273.8 (18.45)
61	176.7 (1.6)	281.8 (4.28)	2746.6 (6.35)	3405.4 (12.19)	165.96 (15.15)	254.49 (17.44)	273.98 (22.98)
62	183 (1.64)	341 (6.09)	2771.4 (6.14)	3408.2 (12.82)	169.02 (21.57)	267.28 (18.98)	273.1 (18.54)
63	168.6 (1.47)	292.7 (5.31)	2651.9 (10.37)	3389.6 (12.74)	186.54 (19.83)	251.57 (15.24)	271.99 (20.08)
64	158.8 (1.42)	318.4 (5.42)	2756.8 (6.14)	3428.4 (13.55)	178 (16.55)	264.8 (19.24)	249.06 (16.75)
65	183.4 (1.73)	303.9 (5.44)	2758.9 (5.85)	3034.9 (14.81)	188.58 (23.85)	259.58 (17.81)	229.05 (13.62)
66	156.4 (1.54)	305 (4.66)	2834.5 (4.11)	3335 (13.61)	170.73 (17.27)	242.67 (17.12)	249.89 (19.46)
67	171.4 (1.55)	374.8 (6.05)	2744.1 (6.17)	3335 (13.27)	188.88 (16.29)	210.19 (14.51)	240.19 (16.5)
68	148 (1.38)	295.9 (3.97)	2810.7 (5.5)	3346.7 (14.17)	166.53 (18.25)	243.66 (16.21)	226.17 (15.67)
69	176.7 (1.68)	277.4 (3.82)	2797.6 (5.37)	3675.4 (10.35)	182.26 (18.39)	266.49 (22.4)	230.4 (14.83)
70	138.5 (1.24)	268.8 (1.99)	2774.9 (5.21)	3213.3 (14.55)	164.2 (15.48)	259.44 (19.24)	239.27 (13.97)

Table B.4: Average clusterings comparison (K-means using Manhattan distance) over the 100 iterations, with standard error of the mean in parentheses for the mouse multi-tissue dataset.

Number of samples	Number of clusters 4	Number of clusters 5	Number of clusters 6	Number of clusters 7	Number of clusters 8	Number of clusters 9	Number of clusters 10
3	2987.3 (126.93)	3527.5 (109.66)	3924.7 (111.14)	4291.1 (168.23)	3737.8 (116.46)	4188.6 (177.57)	4718.6 (96.73)
4	2868.5 (145.62)	3342.3 (140.61)	3848.8 (181.54)	3884.7 (92.62)	3431.4 (96.91)	4120.5 (124.94)	3730.5 (127.96)
5	2469.4 (127.36)	3014.4 (183.24)	3489.1 (145.91)	3169.4 (88.9)	4044.8 (147.89)	4031.6 (127.54)	3969.2 (95.39)
6	2751.9 (147.73)	2679.9 (112.56)	2946.4 (114.79)	3802.7 (68.27)	4011.7 (184.69)	3910 (155.25)	3408.5 (66.28)
7	2572.3 (118.86)	2471.5 (141.13)	2893.8 (125.04)	3471.6 (121.86)	4109.8 (107.39)	3186.6 (124.87)	3816.3 (86.3)

Table B.4 continued from previous page

Number of samples	Number of clusters 4	Number of clusters 5	Number of clusters 6	Number of clusters 7	Number of clusters 8	Number of clusters 9	Number of clusters 10
8	2722.1 (116.87)	2996.2 (117.01)	3171 (131.83)	3753.3 (156.35)	3680.4 (133.43)	2974.8 (101.08)	3711.6 (167.91)
9	2214.3 (162.94)	2639.9 (173.36)	3594.8 (116.17)	3183.9 (114.01)	4015.5 (119.61)	3598.6 (190.19)	3667.9 (105.61)
10	2767.3 (147.64)	2901.7 (63.27)	3452.5 (97.94)	3606.4 (110.36)	2851.2 (128.58)	3446.1 (78.92)	3333.6 (97.79)
11	1397.4 (73.53)	3193.5 (150.43)	2691.1 (144.8)	2610.9 (81.39)	2666 (83.35)	2680 (99.16)	3536.4 (116.18)
12	1855 (118.78)	2573.8 (137.86)	2779.6 (98.78)	3225.9 (132.31)	3274.6 (111.61)	1945.8 (57.72)	3114.3 (96.49)
13	1893.8 (143)	2323.9 (80.52)	3073.2 (164.31)	2898 (147.36)	3626.3 (87.54)	3069.1 (109.98)	2813.5 (104.91)
14	2683.5 (145.29)	3359.7 (132.43)	2761.3 (145.75)	3437.1 (115.84)	3346 (200.52)	2832.3 (144.91)	3993.4 (121.37)
15	1989.9 (145.65)	2979.2 (115.14)	3498.9 (123.28)	2823.6 (98.45)	3949.5 (98.17)	2230 (62.06)	2900.7 (96.7)
16	2072.7 (166.31)	2201.7 (67.71)	2530.8 (135.72)	3440.1 (133.25)	3621.6 (118.99)	3322.6 (137.63)	3026.8 (118.46)
17	1383 (107.04)	2887 (106.27)	2912.6 (135.91)	2596 (88.64)	2973.3 (137.89)	3491.2 (147.49)	2918.8 (145.22)
18	1650.9 (119.9)	2241.4 (108.46)	3145.3 (195.3)	3143.8 (119.69)	3825.3 (102.93)	2216.8 (110.66)	3372.8 (105.85)
19	2900.3 (180.3)	2268.9 (135.97)	2554.5 (120.35)	2190.1 (90.89)	3350.5 (152.93)	2480.8 (75.74)	3555.2 (166.47)
20	2850.3 (171.2)	2169.7 (106.23)	3110.3 (189.11)	2641.8 (64.74)	3330.9 (82.61)	2066.6 (69.15)	3064.3 (130.25)
21	2064.3 (135.32)	1783.7 (102.43)	3274.7 (172.55)	2817.7 (105.41)	3633 (123.28)	2481.3 (132.64)	2748 (117.52)
22	2944.3 (143.8)	2451.4 (122.46)	3067.2 (160.04)	3246.6 (181.19)	3381.6 (90.11)	2701 (100.93)	3022.6 (130.16)
23	1745.6 (133.45)	1867.7 (89.63)	3536.1 (152.89)	2911.2 (113.41)	3536.5 (108.39)	2665.3 (73.8)	2466.1 (51.26)
24	1287.6 (97.66)	2329.7 (82.73)	2858.3 (133.34)	3119.1 (53.41)	3571.6 (84.68)	3204.6 (113.89)	3389.1 (140.17)
25	729.4 (39.02)	2844.9 (101.16)	2790 (134.02)	2756.9 (169.84)	3516.4 (149.76)	2690 (69.74)	2772.9 (62.9)
26	1848.6 (128.75)	1931.1 (93.8)	3018.5 (149.96)	3291.8 (127.12)	3183.1 (111.81)	2699 (121.26)	2701.7 (44.66)
27	2738.4 (156.71)	2399.3 (113.19)	2922.9 (135.53)	2177.5 (84.44)	3012.8 (123.49)	2597.9 (92.42)	2605.4 (90.1)
28	933.7 (121.28)	2264.5 (142.05)	2415.5 (183.76)	2793.5 (106.37)	2797 (108.38)	2124.3 (120.8)	2833.9 (85.66)
29	1482.4 (128.37)	2832 (187.04)	2929 (125)	3731.6 (157.77)	3224.1 (147.08)	2064 (89.89)	3047.6 (131.51)
30	2642.7 (107.65)	2712.1 (101.28)	2486.4 (174.04)	2705.6 (95.5)	3614.7 (87.12)	2187.4 (99.36)	2918.4 (46.58)
31	1938.5 (140.21)	2118.2 (116.67)	2438.5 (117.03)	2642.7 (99.53)	3397.2 (146.41)	1856.3 (87.7)	2850.2 (140.31)
32	1346.7 (138.51)	2044.3 (89.53)	2978.5 (200.46)	2609.2 (119.25)	2902.2 (114.2)	1929.8 (77.9)	2977.7 (136)

Table B.4 continued from previous page

Number of samples	Number of clusters 4	Number of clusters 5	Number of clusters 6	Number of clusters 7	Number of clusters 8	Number of clusters 9	Number of clusters 10
33	1882.9 (163.11)	2023.9 (110.94)	2736.4 (177.57)	2448.7 (141.62)	3528.8 (139.16)	1974.7 (133.49)	3068 (107.86)
34	1831.8 (126.56)	2874.9 (177)	3307.9 (176.56)	2203.2 (102.61)	3457.4 (162.86)	2865.1 (136.15)	2812.7 (93.61)
35	2407.6 (147.62)	2283.6 (145.04)	2013.9 (94.04)	2640.9 (93.82)	3132.5 (178.64)	2202.9 (114.48)	2519.5 (98.62)
36	1423.6 (153.12)	2113.9 (87.01)	2986 (160.3)	1832.1 (101.51)	3702.2 (75.91)	1945.2 (88.26)	2485.2 (85.68)
37	1339.9 (117.95)	2714.8 (170.78)	2613.1 (161.21)	2460.6 (160.1)	3893.3 (120.97)	2461.9 (136.95)	3249.3 (133.61)
38	1898.5 (137.83)	2246.5 (104.72)	1832.3 (165.69)	2599.5 (126.69)	3639.3 (111.1)	2401.2 (125.13)	3020.3 (158.88)
39	959.8 (106.41)	1866.3 (94.33)	1767.1 (118.9)	2365.9 (104.63)	3630.1 (113.67)	2902.7 (103.65)	2920.5 (123.81)
40	2058.1 (222.2)	1875.4 (71.64)	1721.5 (105.1)	2100.4 (64.91)	3589.2 (177.3)	1996.5 (40.65)	3161.7 (120.4)
41	1532.8 (130.58)	2333.4 (128.26)	2280.1 (158.73)	2202.1 (110.1)	3722.4 (77.61)	2820.3 (126.7)	2428.5 (91.73)
42	1077 (116.57)	1724.6 (109.61)	2054.4 (119.12)	2854 (153.64)	3268.9 (73.3)	2143.2 (161.5)	3414.3 (156.5)
43	1996.4 (174.84)	1699.1 (129.95)	2475.1 (138.92)	2609.2 (98.88)	3834.9 (127.33)	2901.9 (126.21)	2527.6 (98.74)
44	2824 (221.96)	2492.4 (53.7)	1988.4 (89.3)	2854.6 (162.09)	3442.6 (106.98)	3121.6 (100.13)	2327.9 (111.39)
45	2113.7 (129.75)	2482.2 (114.43)	2930.4 (172.06)	2505.4 (117.59)	3285.1 (131.76)	2283.5 (111.73)	3037.3 (100.37)
46	2606.4 (160.61)	2104.6 (170.97)	2055.4 (152.54)	2481.5 (144.21)	3317.2 (167.04)	2382.4 (126.18)	2679.8 (104.67)
47	1855.8 (127.77)	1848.1 (145.2)	2053.3 (132.4)	2295.9 (148.47)	3551 (141.73)	1656.2 (89.87)	2821 (116.21)
48	1149 (122.72)	1692.3 (113.57)	2137.5 (115.96)	2478.5 (104.56)	2937.3 (126.46)	2157.7 (72.6)	2786.4 (63.5)
49	1714.8 (152.14)	2332.7 (135.84)	1915.7 (183.23)	2605.1 (94.03)	3015.5 (121.84)	2360.1 (143.79)	2947.3 (125.57)
50	1715.1 (171.83)	2433.5 (122.48)	2856.8 (173.93)	2704.4 (151.86)	3689.4 (92.73)	1686.8 (142)	2566.9 (109.13)
51	702.6 (100.39)	1985.3 (99.19)	1855.8 (149.84)	1924.1 (106.07)	3108.7 (176.28)	2630.7 (100.8)	2702.1 (123.17)
52	987.9 (148.04)	2040 (110.45)	2382.3 (164.25)	2415.9 (107.75)	3734.7 (84.73)	2065.4 (126.21)	2568.2 (48.79)
53	1913.4 (137.2)	2323.2 (117.97)	2323.8 (136.27)	3477.1 (150.7)	3804.7 (148.38)	2005.4 (96.15)	3317.7 (124.36)
54	1233.4 (154.5)	2871.8 (144.33)	2067.1 (128.3)	1688.4 (50.32)	3449.2 (117.78)	1919.5 (139.61)	2003.5 (72.31)
55	1828.1 (164.27)	1714.5 (92.25)	2343.6 (160.26)	2531.4 (150.93)	3383.3 (142.27)	2335.1 (109.19)	2215.5 (85.24)
56	1282.1 (156.57)	1647.4 (107.59)	2378.8 (180.96)	2213 (97.55)	3768.2 (124.74)	1540.8 (88.16)	2451.6 (59.19)
57	1540.7 (149.93)	2297.8 (106.94)	2267.9 (138)	2795.1 (101.51)	2699.8 (133.66)	1950.6 (108.01)	2553.1 (130.19)

Table B.4 continued from previous page

Number of samples	Number of clusters 4	Number of clusters 5	Number of clusters 6	Number of clusters 7	Number of clusters 8	Number of clusters 9	Number of clusters 10
58	1712.3 (181.35)	1851.7 (116.73)	2074.8 (141.35)	1604.4 (64.98)	3536.3 (119.93)	2457.2 (156.18)	2563.8 (112.7)
59	1570.4 (154.03)	2244.3 (111.74)	1044.8 (68.86)	2588.9 (144.9)	3411.9 (100.77)	1633.1 (110.4)	2629.9 (81.31)
60	1653.9 (163.15)	1859.7 (96.12)	2309.6 (158.89)	3166.7 (115.7)	3500.7 (120.04)	2054 (91.94)	2599.2 (139.49)
61	1621.7 (135.63)	1577.3 (91.48)	2362.6 (154.06)	2115 (109.77)	3678.8 (176.51)	1709.4 (98.71)	3005.2 (127.23)
62	1917.4 (172.15)	1498.5 (134.94)	1924 (116.44)	1711.8 (86.73)	3545.7 (134.69)	2652 (104.18)	3043.4 (122.58)
63	1820.5 (148.24)	2154.2 (60.32)	1770.8 (129.23)	2591.4 (164.38)	3839.7 (152.42)	1739.2 (116.39)	2390.1 (121.67)
64	1498.7 (171.47)	2602.8 (101.13)	3360.8 (173.3)	1646.4 (67.06)	3282.7 (113.39)	2404.9 (173.13)	2248.5 (90.71)
65	1923.5 (203.99)	2461.8 (179.34)	1389.6 (138.9)	2609.8 (137.5)	2889.6 (158)	2656.4 (108.8)	2775.8 (137.89)
66	2474.4 (164.97)	2311.1 (132.51)	2761.3 (180.04)	2132.7 (123.78)	4077.7 (104.31)	2097.3 (147.23)	1902.1 (74.15)
67	835.7 (125.77)	2304.5 (141.95)	1825.7 (84.81)	2600.2 (126.59)	3358.3 (164.02)	2641.5 (114.24)	3102.5 (117.86)
68	807.1 (120.02)	2982.1 (112.04)	2849.3 (199.02)	2234.9 (100.06)	2716.2 (167.29)	1600.3 (98.52)	2313.9 (104)
69	2071.9 (122.67)	1875.7 (129.55)	1757.8 (142.24)	2023.3 (157.19)	3646.1 (104.78)	2031.6 (113.99)	2321.4 (73.86)
70	1248 (143.96)	1706.4 (108.73)	2075.8 (161.63)	2092.4 (118.44)	2581.6 (129.18)	1279.8 (125.78)	2901.1 (71.04)