ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ

ΤΜΗΜΑ ΗΛΕΚΤΡΟΝΙΚΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΤΜΗΜΑΤΟΠΟΙΗΣΗ ΓΟΝΙΔΙΩΝ ΒΑΣΙΣΜΕΝΗ ΣΕ ΒΙΟΛΟΓΙΚΗ ΓΝΩΣΗ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΙΩΑΝΝΑ Γ. ΜΠΟΥΖΟΥ

ΕΠΙΒΛΕΠΩΝ: ΜΙΧΑΛΗΣ ΖΕΡΒΑΚΗΣ
ΚΑΘΗΓΗΤΗΣ ΠΟΛΥΤΕΧΝΕΙΟΥ ΚΡΗΤΗΣ (Π.Κ.)

ΧΑΝΙΑ, ΟΚΤΩΒΡΙΟΣ 2010

ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ

ΤΜΗΜΑ ΗΛΕΚΤΡΟΝΙΚΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΤΜΗΜΑΤΟΠΟΙΗΣΗ ΓΟΝΙΔΙΩΝ ΒΑΣΙΣΜΕΝΗ ΣΕ ΒΙΟΛΟΓΙΚΗ ΓΝΩΣΗ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΙΩΑΝΝΑ Γ. ΜΠΟΥΖΟΥ

ΕΠΙΒΛΕΠΩΝ: ΜΙΧΑΛΗΣ ΖΕΡΒΑΚΗΣ
ΚΑΘΗΓΗΤΗΣ ΠΟΛΥΤΕΧΝΕΙΟΥ ΚΡΗΤΗΣ (Π.Κ.)

ΕΓΚΡΙΘΗΚΕ ΑΠΟ ΤΗΝ ΤΡΙΜΕΛΗ ΕΞΕΤΑΣΤΙΚΗ ΕΠΙΤΡΟΠΗ ΤΗΝ 25^{Η} ΟΚΤΩΒΡΙΟΥ 2010.

.....	.....	.....
Μ. ΖΕΡΒΑΚΗΣ	Μ. ΓΑΡΟΦΑΛΑΚΗΣ	Α. ΛΙΑΒΑΣ
ΚΑΘΗΓΗΤΗΣ Π.Κ.	ΚΑΘΗΓΗΤΗΣ Π.Κ.	ΚΑΘΗΓΗΤΗΣ Π.Κ.

ΧΑΝΙΑ, ΟΚΤΩΒΡΙΟΣ 2010

.....
ΙΩΑΝΝΑ Γ. ΜΠΟΥΖΟΥ

ΔΙΠΛΩΜΑΤΟΥΧΟΣ ΗΛΕΚΤΡΟΝΙΚΟΣ ΜΗΧΑΝΙΚΟΣ ΚΑΙ ΜΗΧΑΝΙΚΟΣ ΥΠΟΛΟΓΙΣΤΩΝ Π.Κ.
ibouzou@gmail.com

ΕΥΧΑΡΙΣΤΙΕΣ

Θα ήθελα να ευχαριστήσω θερμά τον καθηγητή του Πολυτεχνείου Κρήτης κ. Μιχάλη Ζερβάκη για την επίβλεψη και τη στήριξη της διπλωματικής μου εργασίας. Κατόπιν θα ήθελα να ευχαριστήσω τον κ. Στέλιο Σφακιανάκη και την κ. Γεωργία Τσιλίκη, διότι με τη συνεργασία τους συνέβαλαν στην ολοκλήρωση αυτής της προσπάθειάς μου.

ΤΜΗΜΑΤΟΠΟΙΗΣΗ ΓΟΝΙΔΙΩΝ ΒΑΣΙΣΜΕΝΗ ΣΕ ΒΙΟΛΟΓΙΚΗ ΓΝΩΣΗ

ΠΕΡΙΛΗΨΗ

Το θέμα της ομαδοποίησης έχει προσελκύσει το επιστημονικό ενδιαφέρον τα τελευταία χρόνια, καθώς μπορεί να εφαρμοστεί σε πολλούς επιστημονικούς τομείς. Αφορά στη δημιουργία διακριτών ομάδων από οντότητες, όπου οντότητες στην ίδια ομάδα έχουν κοινά χαρακτηριστικά, ενώ οντότητες που ανήκουν σε διαφορετικές ομάδες είναι καλά διαχωρίσιμες. Αυτός ο διαχωρισμός των οντοτήτων επιτυγχάνεται με τη χρήση ενός κατάλληλου κριτηρίου, και οι διαχωρίσιμες ομάδες που τελικά προκύπτουν είναι γνωστές στη βιβλιογραφία ως clusters.

Σκοπός της παρούσας διπλωματικής εργασίας είναι να εφαρμόσει τη θεωρία της ομαδοποίησης σε βιολογικές και βιοϊατρικές εφαρμογές. Λόγω ύπαρξης μεγάλου αριθμού γονιδίων και της πολυπλοκότητας των βιολογικών δικτύων, η ομαδοποίηση αποτελεί χρήσιμη τεχνική για την ανάλυση δεδομένων γονιδιακής έκφρασης. Έτσι λοιπόν, στις τμηματικές μεθόδους ομαδοποίησης που εστιάζουμε, επιλέγονται τα κατάλληλα κριτήρια που θα οδηγήσουν σε σωστές προσεγγίσεις ομαδοποίησης. Επιπλέον, καθώς είναι γνωστό πως η ομαδοποίηση είναι πρόβλημα NP-hard, γίνεται επιλογή (και διαμόρφωση όταν κριθεί απαραίτητο) κατάλληλης βελτιστοποιημένης προσέγγισης που οδηγεί σε αποτέλεσμα κοντά στο βέλτιστο. Επίσης, επειδή πολλοί αλγόριθμοι δέχονται ως παράμετρο το πλήθος των ομάδων, η εκτίμηση του βέλτιστου πλήθους αποτελεί κρίσιμο πρόβλημα. Διάφορα κριτήρια εγκυρότητας της προκύπτουσας ομαδοποίησης χρησιμοποιούνται για το παραπάνω πρόβλημα. Ένα ακόμα σημαντικό ζήτημα λόγω ύπαρξης μεγάλου συνόλου δεδομένων, είναι να εξετάσουμε κατά πόσο μοιάζουν (ή να επαληθεύσουμε αν είναι ισοδύναμοι) δύο αλγόριθμοι (όταν ένας είναι πιο απλός και/ή αποτελεσματικότερος του άλλου). Ένα τέτοιο κριτήριο εγκυρότητας χρησιμοποιείται και για την εκτίμηση του κατάλληλου πλήθους των ομάδων. Είναι επίσης σημαντικό να αναφερθεί πως ενσωματώνοντας την πρότερη βιολογική γνώση στη διαδικασία της ομαδοποίησης, προκύπτουν περισσότερο βιολογικής φύσεως ομάδες. Η παραπάνω προσέγγιση θα μπορούσε να υποστηρίξει την ανακάλυψη των ομάδων γονιδίων που έχουν παρόμοιες βιολογικές λειτουργίες. Αυτή η βιολογική γνώση θα μπορούσε να χρησιμοποιηθεί στον αλγόριθμο της ομαδοποίησης και στο

στάδιο της εγκυρότητας της προκύπτουσας ομαδοποίησης. Δυστυχώς, η πρότερη βιολογική γνώση δεν είναι πάντα διαθέσιμη.

Στην παρούσα διπλωματική εργασία, έγινε ο σχεδιασμός και η υλοποίηση κατάλληλων προσεγγίσεων ομαδοποίησης γονιδίων. Μερικές προσεγγίσεις εφαρμόστηκαν σε στατιστική γνώση, δηλαδή σε τρία σύνολα δεδομένων που αφορούν τον καρκίνο του μαστού (Sorlie, Veer και Sotiriou), και άλλες εφαρμόστηκαν σε δύο διαφορετικές μορφές βιολογικής γνώσης, δηλαδή στη διαθέσιμη πληροφορία από την Gene Ontology (GO) και από την Kyoto Encyclopedia of Genes and Genomes (KEGG). Στο τέλος, οι διάφορες ομαδοποιήσεις γονιδίων που προκύπτουν ελέγχονται με τη βοήθεια διάφορων κριτηρίων εγκυρότητας και έτσι προκύπτουν βιολογικής σημασίας συμπεράσματα. Σκοπός είναι να ενισχύσουμε τη στατιστική ανάλυση με τη χρήση διαθέσιμης πρότερης βιολογικής γνώσης. Για να επιτευχθεί αυτό, βιολογικές αποστάσεις, δηλαδή αποστάσεις που υπολογίζονται με βάση τη διαθέσιμη βιολογική γνώση, χρησιμοποιούνται στον αλγόριθμο της ομαδοποίησης και στο στάδιο της εγκυρότητας της προκύπτουσας ομαδοποίησης. Αναφέρουμε πως οι μεθοδολογίες ομαδοποίησης γονιδίων που παρουσιάζονται έχουν υλοποιηθεί στο matlab.

Λέξεις Κλειδιά: ομαδοποίηση γονιδίων, κριτήρια εγκυρότητας, κριτήρια ομοιότητας, βιολογική γνώση, βιολογικές αποστάσεις

CLUSTERING OF GENES BASED ON BIOLOGICAL KNOWLEDGE

ABSTRACT

Cluster analysis has attracted considerable attention the last few years, since can be applied in many scientific fields. It refers to the formation of distinct blocks of objects, where objects within a block have some common characteristics and objects that belong to different blocks are well separated. The separation of the objects is achieved based on an appropriate criterion, while the final distinct blocks are well known as clusters.

The purpose of this thesis, is to accommodate cluster analysis theory to biological and biomedical applications. Because of the large number of genes and the complexity of biological networks, clustering is an useful technique for analysis of gene expression data. The thesis deals with the challenging problem of defining the efficient criteria to guide the selection of the appropriate clustering approaches, focusing on partitional clustering methods. Furthermore, since clustering is a known NP-hard problem, a difficult task is to select (and modify when necessary) the appropriate optimization schemes, that provide a reliable near optimum solution. Also, since many clustering algorithms require the number of clusters as an input parameter, the prediction of the correct number of clusters is a critical problem. Different cluster validity indices have been suggested to address this problem. Additionally, another important issue with current research, where large data sets are so common, is to assess degree of similarity (or verify equivalence) of two clustering algorithms (for example one being a simpler and/or more efficient version of the other). The behavior of such a similarity index can also be used as an indicator of the proper number of clusters in a data set. It is also important to mention that incorporating prior knowledge in the clustering process would generate clusters that are more biologically relevant. Also, this supports the discovery of clusters of genes sharing similar functions. Such a biological knowledge may be used in clustering method and cluster validation. Unfortunately, this sort of prior biological knowledge is not always available.

In this thesis, design and implementation of appropriate gene clustering strategies are achieved. Some clustering approaches are applied on available statistical knowledge, i.e. three data sets concerning breast cancer (Sorlie's, Veer's and Sotiriou's data set), and other on two

types of available biological knowledge, i.e. Gene Ontology (GO) and Kyoto Encyclopedia of Genes and Genomes (KEGG) knowledge. We validate and compare the obtained gene partitions via several measures to make meaningful biological conclusions. The purpose is to enrich the numerical cluster analysis with available prior biological knowledge. To achieve this, biological distances, i.e. distances calculated based on available biological knowledge, are used in clustering method and cluster validation. It is mentioned that the presented gene clustering methodologies have been implemented in matlab.

Keywords: gene clustering, validity measures, similarity indices, biological knowledge, biological distances

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1: STATISTICAL CLUSTER ANALYSIS AND ITS APPLICATIONS

1.1 Introduction

Cluster analysis is a basic human mental activity and has an important role in research developed across a wide variety of communities. Cluster analysis is defined as a way to create groups or objects, or clusters, in such a way that objects in one cluster are very similar and objects in different clusters are quite distinct. It has many alternative names differing from one discipline to another. In biology and ecology, cluster analysis is more often known as numerical taxonomy. Researchers in computational intelligence and machine learning are more likely to use the terms unsupervised learning or learning without a teacher. In social science, typological analysis is preferred, while in graph theory, partition is usually employed. This diversity reflects the important position of clustering in scientific research. On the other hand, it causes confusion because of the differing terminologies and goals. Frequently, similar theories or algorithms are redeveloped several times in different disciplines due to the lack of good communication, which causes unnecessary burdens and wastes time.

In this chapter, we introduce the basic concepts of cluster analysis. The type of data is a major factor to consider in choosing an appropriate clustering algorithm. A similarity measure[1] or distance (dissimilarity measure[2]) is used to quantitatively describe the similarity or dissimilarity of two clusters without which no meaningful cluster analysis is possible. Generally, clustering algorithms can be classified to two categories: hard clustering algorithms and fuzzy clustering algorithms. Unlike hard clustering algorithms, which require that each data point of the data set belong to one and only one cluster, fuzzy clustering algorithms allow a data point to belong to two or more clusters with different probabilities. Furthermore, we describe the two most significant clustering methods, i.e. hierarchical and partitional. Finally, this chapter introduces the application of clustering to gene expression[3] data.

1.2 Cluster Analysis

One of the most important of data analysis activities is to classify or group data into a set of categories or clusters through clustering algorithms. In particular, clustering algorithms partition data objects (patterns, entities, instances, observances, units) into a certain number of clusters (groups, subsets, or categories). Data objects that are classified in the same group should display similar properties based on some criteria. Unfortunately, there is no universally agreed upon and precise definition of the term cluster. In one approach a cluster is defined as a set of entities which are alike, and entities from different clusters are not alike. Alternatively, a cluster is an aggregate of points in the test space such that the distance between any two points in the cluster is less than the distance between any point in the cluster and any point not in it. Also, clusters may be described as continuous regions of this space (d-dimensional feature space) containing a relatively high density of points, separated from other such regions containing a relatively low density of points. Generally, classification systems are either supervised or unsupervised, de-

---

1 It measures how much two objects resemble each other.

2 It measures how far away two objects are from each other.

3 Gene expression is the process by which the heritable information in a gene, which is the sequence of DNA base pairs, is made into a functional gene product, such as protein or RNA.

pending on whether they assign new data objects to one of a finite number of discrete supervised classes or unsupervised categories. Also, there are some other classification systems which are semi-supervised.

In supervised classification, the mapping from a set of input data vectors, denoted as $\mathbf{x} \in R^d$, where d is the input space dimensionality, to a finite set of discrete class labels, represented as $y \in 1,\ldots,C$, where C is the total number of class types, is modeled in terms of some mathematical function $y = y(\mathbf{x},\mathbf{w})$, where $\mathbf{w}$ is a vector of adjustable parameters. The values of these parameters are determined (optimized) by an inductive learning algorithm (also termed inducer), whose aim is to minimize an empirical risk functional (related to an inductive principle) on a finite data set of input - output examples, $(x_i, y_i), i = 1,\ldots,N$, where N is the finite cardinality of the available representative data set. When the inducer reaches convergence or terminates, an induced classifier is generated.

In unsupervised classification, also called clustering or exploratory data analysis, no labeled data are available. Cluster analysis or clustering, which is a core task in data mining, is the assignment of a set of observations into subsets (called clusters) so that observations in the same cluster are similar in some sense. The goal of clustering is to separate a finite, unlabeled data set into a finite and discrete set of "natural", hidden data structures. It is clear that a direct reason for unsupervised clustering comes from the requirement of exploring the unknown natures of the data that are integrated with little or no prior information. Consider, for example, disease diagnosis and treatment in clinics. For a particular type of disease, there may exist several unknown subtypes that exhibit similar morphological appearances while responding differently to the same therapy. In this context, cluster analysis with gene expression data, provides a promising method to uncover the subtypes and thereby determine the corresponding therapies. Sometimes, the process of labeling data samples may become extremely expensive and time consuming, which also makes clustering a good choice considering the great savings in both cost and time. In addition, cluster analysis provides a compressed representation of the data and is useful in large - scale data analysis.

Clustering is a well known problem, and there are many algorithms for cluster analysis in the literature. Cluster analysis emphasizes both internal homogeneity and external separation. The

performance of a clustering algorithm would be improved, if the algorithm could either minimize intracluster distance or maximize intercluster distance. Cluster analysis aims to seek a partition of the data in which data objects in the same clusters are homogenous while data objects in different groups are well separated. This homogeneity and separation are evaluated through the criterion functions. As pointed out by the authors in [1], in cluster analysis a group of objects is split up into a number of more or less homogeneous subgroups based on a subjectively chosen measure of similarity, such that the similarity between objects within a subgroup is larger than the similarity between objects belonging to different subgroups. Moreover, a different clustering criterion or a different clustering algorithm or the same algorithm but with different selection of parameters, may cause completely different clustering results. For instance, human beings may be classified based on their ethnicity, region, age, socioeconomic status, education, career, hobby, weight and height, favorite food, dressing style, and so on. Apparently, different clustering criteria may assign a specific individual to very different groups and therefore produce different partitions. However, there is absolutely no way to determine which criterion is the best in general. As a matter of fact, each criterion has its own appropriate use corresponding to particular occasions, although some of them may be applied to wider situations than others.

Finally, in semi-supervised classification, a small amount of knowledge is available concerning either pairwise (must-link or cannot-link) constraints between data items or class labels for some items. Instead of simply using this knowledge for the external validation of the results of clustering, one can imagine letting it “guide” or “adjust” the clustering process, i.e. provide a limited form of supervision. The resulting approach is called semi-supervised clustering. We also consider that the available knowledge is too far from being representative of a target classification of the items, so that supervised learning is not possible. Note that class labels can always be translated into pairwise constraints for the labeled data items and, reciprocally, by using consistent pairwise constraints for some items one can obtain groups of items that should belong to a same cluster.

1.3 Methods of Clustering

At first, we mention some criteria that provide significant distinction between clustering methods and can help selecting appropriate candidate methods for one’s problem:

Objective of Clustering

Many methods aim at finding a single partition of the collection of items into clusters. However, obtaining a hierarchy of clusters can provide more flexibility. A partition of the data can be obtained from a hierarchy by cutting the tree of clusters at some level.

Nature of the Data Items

Most clustering methods were developed for numerical data, but some can deal with categorical data or with both.

Nature of the Available Information

Many methods rely on rich representations of the data (e.g. vectorial) that let one define prototypes, data distributions, multidimensional intervals, etc., besides computing (dis)similarities. Other methods only require the evaluation of pairwise (dis)similarities between data items, while imposing fewer restrictions on the data. These methods usually have a higher computational complexity.

Nature of the Clusters

The degree of membership of a data item to a cluster is either in [0, 1] if the clusters are fuzzy or in {0, 1} if the clusters are crisp. For fuzzy clusters, data items can belong to some degree to several clusters that don't have hierarchical relations with each other. This distinction between fuzzy and crisp can concern both the clustering mechanisms and their results. Crisp clusters can always be obtained from fuzzy clusters.

Clustering Criterion

Clusters can be seen either as distant compact sets or as dense sets separated by low density regions. Unlike density, compactness usually has strong implications on the shape of the clusters, so methods that focus on compactness should be distinguished from methods that focus on the density.

Several taxonomies of clustering methods were suggested in [2], [3] or [4]. But given the high number and the strong diversity of the existing clustering methods, it is probably impossible to obtain a categorization that is both meaningful and complete. By focusing on some of the discriminating criteria just mentioned, we put forward the simplified taxonomy shown below, inspired by the one suggested in [4]. So, some possible methods of clustering are:

Clustering procedures yield a data description in terms of clusters or groups of data points that possess strong internal similarities.

Distance-Based Clustering

Two or more objects belong to the same cluster if they are "close" according to a given distance (in this case geometrical distance). An example of such methods is k-medoids. The most common realization of k-medoids clustering is the Partitioning Around Medoids (PAM) algorithm [3].

Conceptual Clustering

Two or more objects belong to the same cluster if this one defines a concept common to all these objects. In other words, objects are grouped according to their fit to descriptive concepts, not according to simple similarity measures. Conceptual clustering builds a structure out of the data incrementally by trying to subdivide a group of observations into subclasses. The result is a hierarchical structure known as the concept hierarchy. Each node in the hierarchy subsumes all the nodes underneath it, with the whole data set at the root of the hierarchy tree. Examples of such methods are a conceptual clustering algorithm known as CLUSTER/2 [56] and an improved conceptual clustering algorithm known as ITERATE [57].

Divisive or Partitional Clustering

These methods start with each point as part of a random or guessed cluster and iteratively move points between clusters until some local minimum is found with respect to some distance metric between each point and the center of the cluster it belongs to. Partitional clustering assigns a set of data points into k clusters without any hierarchical structure. This process usually accompanies the optimization of a criterion function. More specifically, given a set of points $x_i \in R^d$, $i = 1, \ldots, N$, partitional clustering algorithms aim to organize them into k clusters $\{C_1, ..., C_K\}$ while maximizing or minimizing a prespecified criterion function J. In principle, the optimal partition, based on the criterion function J, can be found by enumerating all possibilities. However, this brute force method is infeasible in practice due to the extremely expensive computation. Even for a small-scale clustering problem, simple enumeration is impossible. Therefore, heuristic algorithms seek approximate solutions. One of the widely used iterative op-

timization methods, the k-means algorithm is based on the sum-of-squared-error criterion. In this study, different approaches of the k-means algorithm are implemented.

Hierarchical Clustering

Hierarchical clustering groups data with a sequence of nested partitions, either from singleton clusters to a cluster including all individuals or vice versa. The former is known as agglomerative hierarchical clustering, and the latter is called divisive hierarchical clustering. Both agglomerative and divisive clustering methods organize data into the hierarchical structure based on the proximity matrix. The results of hierarchical clustering are usually depicted by a binary tree or dendrogram, as depicted in Figure 1.

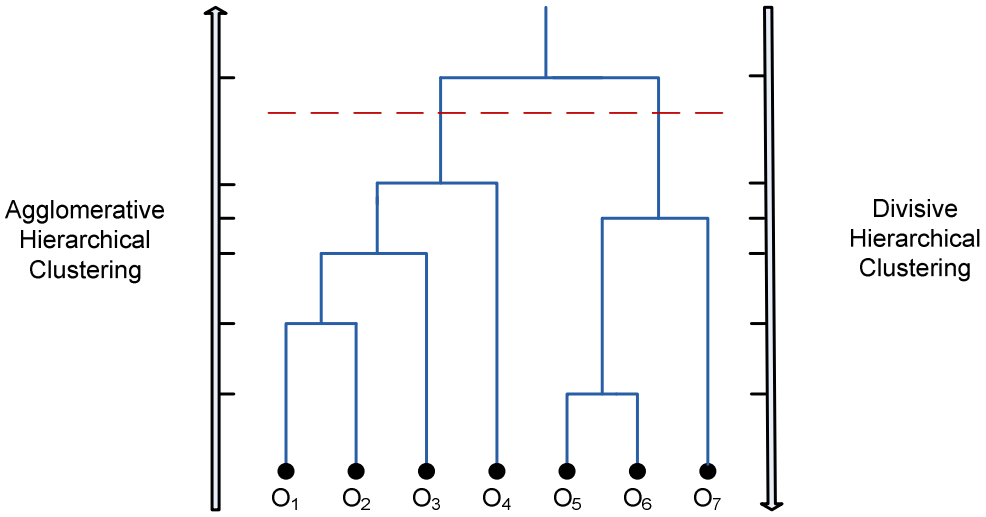


Figure 1 : Example of a dendrogram from hierarchical clustering. The clustering direction for the divisive hierarchical clustering is opposite to that of the agglomerative hierarchical clustering. Two clusters are obtained by cutting the dendrogram at an appropriate level.

These methods start with each point being considered as a cluster and recursively combine pairs of clusters (subsequently updating the intercluster distances) until all points are part of one hierarchically constructed cluster. Hierarchical clustering groups data with a sequence of nested partitions, either from singleton cluster to a cluster including all individuals or vice versa. The results of hierarchical clustering are usually depicted by a binary tree or dendrogram, as depicted in Figure 1. The root node of the dendrogram represents the whole data set, and each leaf node is regarded as a data point. The intermediate nodes thus describe the extent to which the objects are proximal to each other and the height of the dendrogram usually expresses the distance between each pair of data points or clusters, or a data point and a cluster. The ultimate clustering results

can be obtained by cutting the dendrogram at different levels. This representation provides very informative descriptions and a visualization of the potential data clustering structures, especially when real hierarchical relations exist in the data.

Compared with agglomerative methods, divisive methods need to consider $2^{N-1}-1$ possible two-subset divisions for a cluster with N data points, which is very computationally intensive even for small-scale data sets [11]. Therefore, agglomerative methods are more widely used. The major disadvantage of divisive methods is their computational complexity, which is at least $O\left(N^2\right)$ and cannot meet the requirement for dealing with large-scale data sets in data mining and other tasks in recent years [11]. Also, common criticisms of classical hierarchical clustering algorithms focus on their lack of robustness and their sensitivity to noise and outliers. Once an object is assigned to a cluster, it will not be considered again, which means that hierarchical clustering algorithms are not capable of correcting possible previous misclassification. As a result, many new clustering methods with hierarchical cluster results have appeared and have greatly improved the clustering performance.

Graph Theoretic Methods

The concepts and properties of graph theory make it very convenient to describe clustering problems by means of graphs. These methods are partitioning methods that partition the space into subgraphs with respect to some geometric properties. The authors in [49] provide a detailed description and discussion of hierarchical clustering from the point of view of graph theory. More discussion of graph theory in clustering can be found in [50]. Examples of such methods are a k-nearest-neighbor graph-based algorithm, Chameleon [51], and the algorithm CLICK (Clustering Identification via Connectivity Kernels) [52]. Also, Bayesian networks belong to these methods of clustering, since it is a probabilistic graphical model that represents a set of random variables and their conditional dependences via a directed acyclic graph (DAG). For example, a Bayesian network could represent the probabilistic relationships between diseases and symptoms. Given symptoms, the network can be used to compute the probabilities of the presence of various diseases. More information about Bayesian networks can be found in [58].

Fuzzy Clustering

So far, the clustering techniques we have discussed are referred, to as hard or crisp clustering, which means that each data object is assigned to only one cluster. For fuzzy clustering, this restriction is relaxed, and the object can belong to all of the clusters with a certain degree of membership. This is particularly useful when the boundaries between clusters are ambiguous and not well separated. Examples of such methods are Fuzzy C-Means (FCM) [53], the Possibilistic Fuzzy C-Means (PFCM) model proposed by the authors in [54] and the Mountain Method (MM) [55].

1.4 Procedure of Cluster Analysis

The procedure of cluster analysis consists of four basic steps, shown in Figure 2. The red arrows in Figure 2 point to the novel work in this thesis, i.e. incorporating prior knowledge in the clustering process, especially in clustering algorithm and in cluster validation. More details about this work are discussed in Sections 1.7. In the following, we present these steps.

Feature Selection or Extraction

Feature selection chooses distinguishing features from a set of candidates, while feature extraction utilizes some transformations to generate useful and novel features from the original ones. Clearly, feature extraction is potentially capable of producing features that could be of better use in uncovering the data structure. However, feature extraction may generate features that are not physically interpretable. On the contrary, feature selection assures the retention of the original physical meaning of the selected features. In the literature, these two terms sometimes are used interchangeably without further identifying the difference. Both feature selection and feature extraction are very important to the effectiveness of clustering applications. Elegant selection or generation of salient features can greatly decrease the storage requirement and measurement cost, simplify the subsequent design process, and facilitate the understanding of the data. Generally, ideal features should be of use in distinguishing patterns belonging to different clusters, immune to noise, and easy to obtain and interpret. Feature selection is more often used in the context of supervised classification with class labels available.

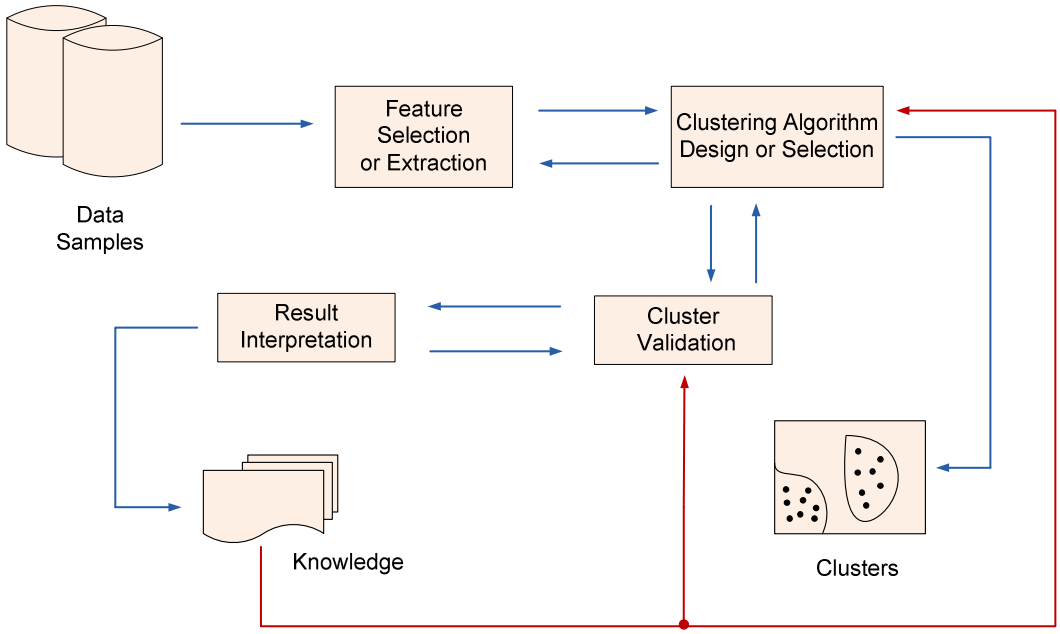


Figure 2 : Clustering procedure. The basic process of cluster analysis consists of four steps with a feedback pathway. These steps are closely related to each other and determine the derived clusters. The red arrows point to the novel work in this thesis.

Clustering Algorithm Design or Selection

This step usually consists of determining an appropriate proximity measure and constructing a criterion function. Intuitively, data objects are grouped into different clusters according to whether they resemble one another or not. The obtained clusters are dependent on the selection of the criterion function. The subjectivity of cluster analysis is thus inescapable. There is no universal clustering algorithm to solve all problems. It is important to carefully investigate the characteristics of a problem in order to select or design an appropriate clustering strategy. Clustering algorithms that are developed to solve a particular problem in a specialized field usually make assumptions in favor of the application of interest. For example, the k-means algorithm is based on the Euclidean measure and hence tends to generate hyperspherical clusters. However, if the real clusters are in other geometric forms, k-means may no longer be effective, and we need to resort to other schemes. Similar considerations must be kept in mind for mixture - model clustering, in which data are assumed to come from some specific models that are already known in advance [11].

Cluster Validation

Given a data set, each clustering algorithm can always produce a partition whether or not there really exists a particular structure in the data. Moreover, different clustering approaches usually lead to different clusters of data, and even for the same algorithm, the selection of a parameter or the presentation order of input patterns may affect the final results. Therefore, effective evaluation standards and criteria are critically important to provide users with a degree of confidence for the clustering results. An unsupervised learning procedure is usually more difficult to assess than a supervised one. The procedure for evaluating the results of a clustering algorithm is known as cluster validation. Although a clustering structure resulting from a certain algorithm could be assessed by domain knowledge and expert experience, cluster validity emphasizes the evaluation of the clustering result in an objective and quantitative way, which is usually statistically based. These assessments should be objective and have no preferences to any algorithm. They should be able to provide meaningful insights in answering questions like how many clusters are hidden in the data, whether the clusters obtained are meaningful from a particular point of view or just artifacts of the algorithms, or why we choose one algorithm instead of another. The first question concerns the cluster tendency of the data and should in principle be answered before attempting to perform clustering, using specific statistical tests. Unfortunately, such tests are not always very helpful and require the formulation of specific test hypotheses. The other questions concern the analysis of cluster validity and can only be answered after application of clustering method to the data.

Generally, there are three types of validation procedures: external indices, internal indices, and relative indices [11]. External indices are based on some prespecified structure, which is the reflection of prior information on the data and is used as a standard to validate the clustering solutions. External validation can only be performed when prior knowledge of the problem is available. The prior knowledge may concern general characteristics of the clusters (e.g. expected compactness) or relations between specific items (e.g. items A and B should belong to a same cluster and item C to a different one). Sometimes this knowledge is confirmatory but not prescriptive. Internal tests are not dependent on external information (prior knowledge). Instead, they examine the clustering structure directly from the original data. Internal validation is based on an evaluation of the “agreement” between the data and the partition. For fuzzy partitional methods, internal validity indices should take into account both the data items and the membership

degrees resulting from clustering. Relative criteria emphasize the comparison of different clustering structures in order to provide a reference to decide which one may best reveal the characteristics of the objects. Relative comparisons are often employed for selecting good values for important parameters, such as the number of clusters.

It is important to mention that in this study, internal indices, for example C-index, Silhouette index, Dunn index and Goodman-Kruskal index, have been applied. We discuss in detail about them in Chapter 3(Section 3.6). Also, external indices, for example Rand index, Hubert index and corrected Rand index, have been applied too. We discuss in detail about them in Chapter 3(Section 3.11). The above indices are the most common used for estimating the number of clusters in a dataset and evaluating the results of a clustering algorithm in gene expression data analysis [8], [16], [21], [17], [44] and [45].

Result Interpretation

The ultimate goal of clustering is to provide users with meaningful insights from the original data so that they can develop a clear understanding of the data and therefore effectively solve the problems encountered. A set of clusters is not itself a finished result but only a possible outline. Consequently, further analyses and experiments may be required.

It is interesting to observe that the flow chart in Figure 2 also includes a feedback pathway. Cluster analysis is not an one-shot process. In many circumstances, clustering requires a series of trials and repetitions. Moreover, there are no universally effective criteria to guide the selection of features and clustering schemes. Validation criteria provide some insights into the quality of clustering solutions, but even choosing an appropriate criterion is a demanding problem. Since clustering is a known NP-hard problem [11], most approaches use the alternative optimization schemes in order to find a local optimum solution of their criterion function.

Finally, it is also important to mention that incorporating prior knowledge in the clustering process would generate clusters that are more biologically relevant. Also, this supports the discovery of clusters of genes sharing similar functions. Such clusters may indicate regulatory pathways, which could be significantly relevant to specific phenotypes or physiological conditions. Such a biological knowledge may be used in clustering method and cluster validation. Red

arrows in Figure 2 represent this work. Unfortunately, this sort of prior biological knowledge is not always available.

1.5 Clustering Applications

Clustering has been applied in a wide variety of fields, as illustrated below with a number of typical applications.

1. *Engineering* (computational intelligence, machine learning, pattern recognition, mechanical engineering, electrical engineering). Typical applications of clustering in engineering range from biometric recognition and speech recognition, to radar signal analysis, information compression, and noise removal.

2. *Computer sciences*. We have seen more and more applications of clustering in web mining, spatial database analysis, information retrieval, textual document collection, and image segmentation.

3. *Life and medical sciences* (genetics, biology, microbiology, paleontology, psychiatry, clinic, phylogeny, pathology). These areas consist of the major applications of clustering in its early stage and will continue to be one of the main playing fields for clustering algorithms. Important applications include taxonomy definition, gene and protein function identification, disease diagnosis and treatment, and so on.

4. *Astronomy and earth sciences* (geography, geology, remote sensing). Clustering can be used to classify stars and planets, investigate land formations, partition regions and cities, and study river and mountain systems.

5. *Social sciences* (sociology, psychology, archeology, anthropology, education). Interesting applications can be found in behavior pattern analysis, relation identification among different cultures, construction of evolutionary history of languages, analysis of social networks, archeological finding and artifact classification, and the study of criminal psychology.

6. *Economics* (marketing, business). Applications in customer characteristics and purchasing pattern recognition, grouping of firms, and stock trend analysis all benefit from the use of cluster analysis.

1.6 Gene Clustering

Over the past few years DNA microarrays[4] have become a key tool in functional genomics. They allow monitoring the expression of thousands of genes in parallel over many experimental conditions (e.g. tissue types, growth environments). This technology enables researchers to collect significant amounts of data, which need to be analyzed to discover functional relationships between genes or samples. The results from a single experiment are generally presented in the form of a data matrix in which rows represent genes and columns represent conditions. Each entry in the data matrix is a measure of the expression level of a particular gene under a specific condition. A central step in the analysis of DNA microarray data is the identification of groups of genes and/or conditions that exhibit similar expression patterns. Clustering is a fundamental approach to classifying expression patterns for biological and biomedical applications. The main assumption is that genes that are contained in a particular functional pathway[5] should be co-regulated and therefore should exhibit similar patterns of expression [7].

DNA microarrays offer a global view of the levels of activity of many genes simultaneously. In a typical gene expression data set, the number of genes is usually such larger than the number of experiments. Even a simple organism like yeast[6] has approximately six thousand genes. It is estimated that humans have approximately thirty thousand to forty thousand genes. Because of

---

[4] A DNA microarray is a multiplex technology used in molecular biology. It consists of an arrayed series of thousands of microscopic spots of DNA oligonucleotides, called features, each containing picomoles (10−12 moles) of a specific DNA sequence, known as probes (or reporters). Since an array can contain tens of thousands of probes, a microarray experiment can accomplish many genetic tests in parallel.

[5] A genetic pathway is the set of interactions occurring between a group of genes who depend on each other's individual functions in order to make the aggregate function of the network available to the cell.

[6] Yeasts are eukaryotic micro-organisms.

the large number of genes and the complexity of biological networks, clustering is a useful exploratory technique for analysis of gene expression data. Clustering has been a useful data-mining tool since early days, for discovering similar expression patterns without prior knowledge. Many clustering algorithms have been proposed for the analysis of gene expression data. The overflowing clustering techniques can further confuse biologists, due to the lack of adequate standards for cluster validity. Clustering algorithms attempt to partition the genes into groups exhibiting similar patterns of variation in expression level. In an attempt to understand complicated biological systems, large amounts of gene expression data have been generated by researchers. Given the same data set, different clustering algorithms can potentially generate very different clusters: the number of clusters and their constituents.

Microarray experiments have been widely used to screen biological activities and cellular changes under different conditions at molecular level. Its ability to simultaneously monitor expression changes of thousands of genes has acquired its popularity but, at the same time, posed many challenging statistical and computational problems. Gene clustering problem is one of them. The purpose of gene clustering is to search for groups of genes with similar expression patterns, which likely have related biological functions or interactions. The complex structure of microarray data, the local optimization or the right choice of the clustering parameters that influence the assignment of genes to clusters are some of the fundamental problems in gene expression clustering. Also, especially in hierarchical clustering approaches it is difficult to identify the “borderline” patterns, i.e. genes with expression profiles that lie between two or more clusters.

A biologist with a gene expression data set is faced with the problem of choosing an appropriate clustering algorithm for his or her data set. The success of clustering algorithms is assessed by visual inspection using biological knowledge. Also, incorporating prior knowledge in the clustering process would help tease out noise and generate clusters that are more refined and biologically relevant. It provides an alternative to avoid all the aforementioned difficulties. So, with the utilization of knowledge background (i.e. knowledge about the function of genes) it is also possible to solve the “borderline” problem, and make the interpretation of the final clustering result more natural. In this study, we presented a novel clustering approach that utilizes information about the functional classification of genes in order to achieve a more knowledgeable and

more naturally interpretable clustering arrangement of the genes. Unfortunately, this sort of prior biological knowledge is not always available.

There is a variety of available sources of biological knowledge. In this study we take advantage of the Gene Ontology (GO) and the Kyoto Encyclopedia of Genes and Genomes (KEGG) based on the Biological Pathways (PWs). More details about the above sources are discussed in Chapter 2. So, we incorporate the available prior knowledge from the above sources in the clustering process. Especially we use biological knowledge in the used clustering algorithm and in cluster validation.

Then the biological meanings of the results are therefore interpreted manually and this work can be time-consuming for large-scale data. Intuitively, a clustering has possible biological significance if genes in the same cluster tend to have similar expression levels in additional experiments that were not used to form the clusters. Because co-expressed genes are likely to share the same biological function, cluster analysis of gene expression profiles has been applied for gene function discovery. It has observed that genes with the same function or involved in the same biological process are likely to co-express, hence clustering gene expression profiles provides a means for gene function prediction.

Different clustering algorithms optimize different objective functions or criteria based on a biological network. Partitional clustering methods such as k-means assign each gene to a single cluster. However, these methods do not provide information about the influence of a given gene for the overall shape of clusters. On the other hand, fuzzy partitioning method, fuzzy c-means (FCM) attributes cluster membership values to genes. Fuzzy clustering is a convenient method to select genes exhibiting tight association to given clusters. In addition to the specification of the number c of clusters in the data set, the FCM method requires to choose m, the fuzziness parameter. Thus, a major problem in applying the FCM method is the choice of the fuzziness parameter m. By setting threshold levels for the membership values, genes which are tightly associated to a given cluster can be selected.

Ideally, we would like to be able to compare proposed clusterings having different numbers of clusters. Unfortunately, determining the correct number of clusters in real data is a long-standing and very difficult problem. The best way to cluster gene expression data is to use more

than one clustering algorithms and compare the results so as to choose the best clustering algorithm. Also, clustering algorithms that may give different results based on different initial conditions should be run several times to find the best solution. Gene expression data clustering is a powerful tool for arranging genes according to similarity in their expression patterns. Cluster analysis is also the first step in analyzing gene expression data. Many traditional clustering algorithms such as k-means can be used to cluster gene expression data.

To sum up, recent advances in DNA microarray technology, also known as gene chips, allow measuring the expression of thousands of genes in parallel and under multiple experimental conditions. This technology is having a significant impact on genomic and post-genomic studies. Disease diagnosis, drug discovery and toxicological research benefit from the use of microarray technology. A main step in the analysis of gene expression data is the detection of samples or genes with similar expression patterns. A number of data mining techniques have been applied to the analysis of gene expression data. Clustering is a fundamental approach to gene expression knowledge discovery. Solutions for the systematic evaluation of the quality of the clusters have been recently proposed. Moreover, the prediction of the correct number of clusters is a critical problem in unsupervised classification problems. Many clustering algorithms require the number of clusters given as an input parameter. Different cluster validity indices have been suggested to address this problem. A cluster validity index indicates the quality of a resulting clustering process. Thus, the clustering partition that optimizes the validity index under consideration is chosen as the best partition. There are several cluster validity techniques for gene expression data analysis. Normalization and validity aggregation strategies are also proposed to improve the prediction of the correct number of clusters in a data set. Also, incorporating prior knowledge in the clustering process leads to clusters that are more refined and biologically relevant.

1.7 Structure and Contribution of This Thesis

In this thesis, one clustering and four validation algorithms are applied to three breast cancer datasets. The combination of these methods may be reliably used for the estimation of the number of clusters and the validation of clustering results. The results show that this software tool can support biomedical knowledge discovery and healthcare applications. We implement the clustering of genes using the hard c-means algorithm and several validity measures (C-index,

Goodman-Kruskal index, Dunn index and Silhouette index) to estimate the number of clusters. Appropriate normalization and weighted voting techniques are used to improve the prediction of the number of clusters. The gene expression values of three data sets concerning the breast cancer (Sorlie's data set, Veer's data set and Sotiriou's data set) are considered, from which the statistical clusters are obtained through the clustering procedure. Furthermore, we annotate the genes to GO and the biological clusters are obtained through the clustering procedure, which now uses only the biological knowledge about the available genes. This knowledge comes from the GO hierarchies, discussed in Chapter 2 in detail. Another approach to obtain the biological clusters, is through the clustering procedure which uses only the biological knowledge in terms of pathways (PWs) from Kyoto Encyclopedia of Genes and Genomes (KEGG). The main idea of this approach is that if two genes take part in at least one common pathway, they should both belong to the same cluster. The validity of this choice is explained in Section 3.12.

The purpose of this thesis is to examine to which extent is possible to obtain biological clusters that converge to statistical clusters. As mentioned, the statistical clusters are obtained through the use of statistical knowledge, i.e. the three available datasets. Note that the biological clusters come from the biological knowledge, i.e. from the GO hierarchies or KEGG. There are several measures to compare different partitions (e.g. Rand similarity index, Hubert similarity index, Rand index after correction for agreement due to chance), some of which have been implemented. Figure 3 summarizes all applied distance metrics, validity and similarity indices, while Figure 4 illustrates the overall contribution presenting the examined gene clustering methodologies in this thesis. The various terms shown in these figures are explained in detail throughout the thesis.

It is worthy noted, that except the statistical cluster analysis using the gene expression values from the three data sets, the contribution of this thesis concerns the incorporation of prior biological knowledge from the GO in clustering procedure and in cluster validation. The results show that the utilization of GO biological knowledge or statistical knowledge leads to clusters that converge adequately with the clusters obtained from KEGG knowledge. Hence, it is possible to design different algorithm approaches, able to use multiple alternative resources and provide reliable gene partitions. Finally, it is shown to what extend is possible to influent the statistical methods with biological knowledge, to obtain results with biological meaning.

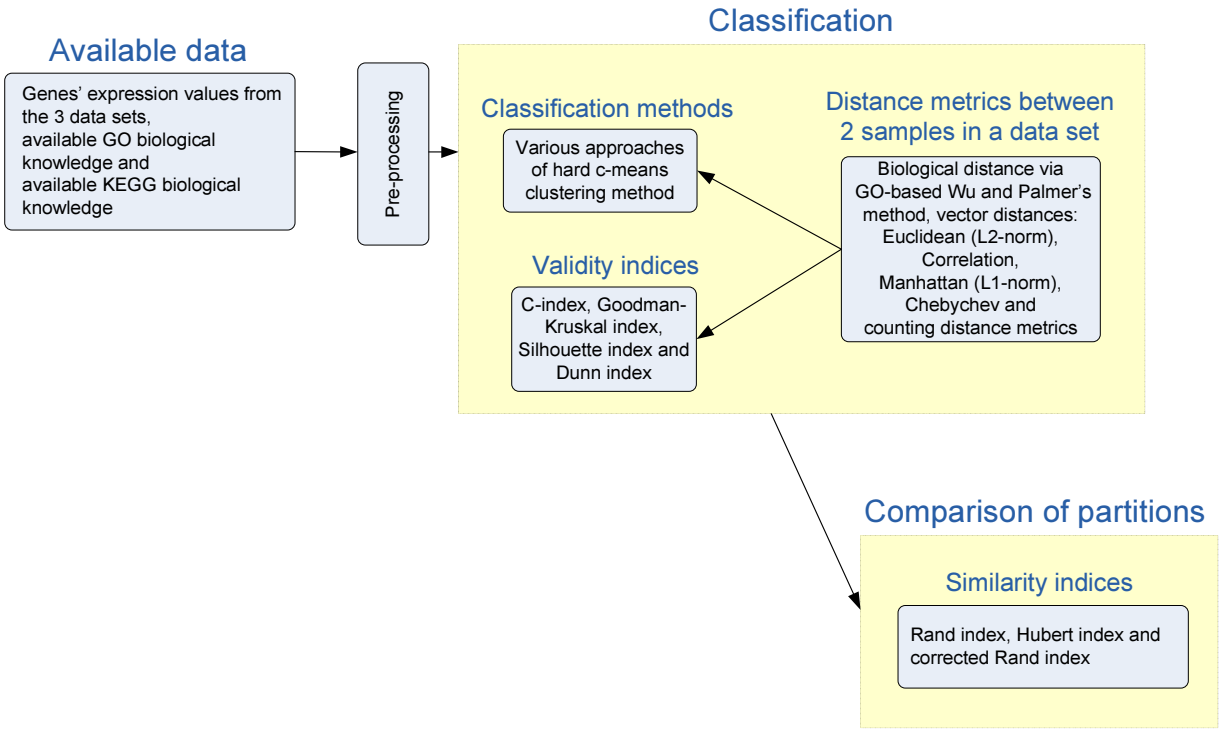


Figure 3 : All distance metrics, validity and similarity indices applied.

To summarize, this work is organized to four chapters covering the following subjects:

- available sources of biological knowledge (GO, KEGG)
- gene clustering methodologies
- cluster validity assessment (validity measures, normalization and weighted voting techniques, similarity indices)
- results interpretation
- motivation for further research

In **Chapter 2** two types of up to date available sources of biological knowledge (the GO and the KEGG) are presented. We introduce their structure and the main available tools that use the data provided by these sources.

In **Chapter 3** we present the implemented gene clustering methodologies and data- and knowledge-driven cluster validity assessment system. Normalization and weighted voting techniques are used to improve the prediction of the number of clusters. Also, we discuss about sev-

eral measures to compare different partitions, some of which have been implemented and evaluated in Chapter 4.

In **Chapter 4** we present a comparative experimental evaluation of the implemented gene clustering methodologies, aiming at illustrating their advantages and disadvantages. We compare and interpret the obtained results and we make meaningful biological conclusions.

In **Chapter 5** we summarize in brief the findings in Chapter 4. We also introduce some novel ideas to motivate further research. We suggest some guidelines about the implemented gene clustering methodologies, which might lead to better results and then, to more meaningful biological conclusions.

1.8 Summary

Cluster analysis aims to provide a partition of the data where data objects in the same clusters are homogenous, while data objects in different groups are well separated. The procedure of cluster analysis consists of four basic steps: feature selection or extraction, clustering algorithm design or selection, cluster validation and result interpretation. There are no universally effective criteria to guide the selection of the appropriate clustering method for a specific problem. The two most significant clustering methods presented in this chapter, are the hierarchical and the partitional clustering method. Cluster analysis is not an one-shot process. Since clustering is a known NP-hard problem, most approaches use more efficient optimization schemes to find a local optimum solution. In biological and biomedical applications, clustering algorithms attempt to partition the genes into groups exhibiting similar patterns of variation in expression level. It has been a useful data-mining tool since early days, for discovering similar expression patterns without prior knowledge. There are several cluster validity techniques for gene expression data analysis. Normalization and validity aggregation strategies are also proposed to improve the prediction of the correct number of clusters in a data set. Also, incorporating prior knowledge in the gene clustering process would lead to a more knowledgeable and more naturally interpretable clustering arrangement of the genes.

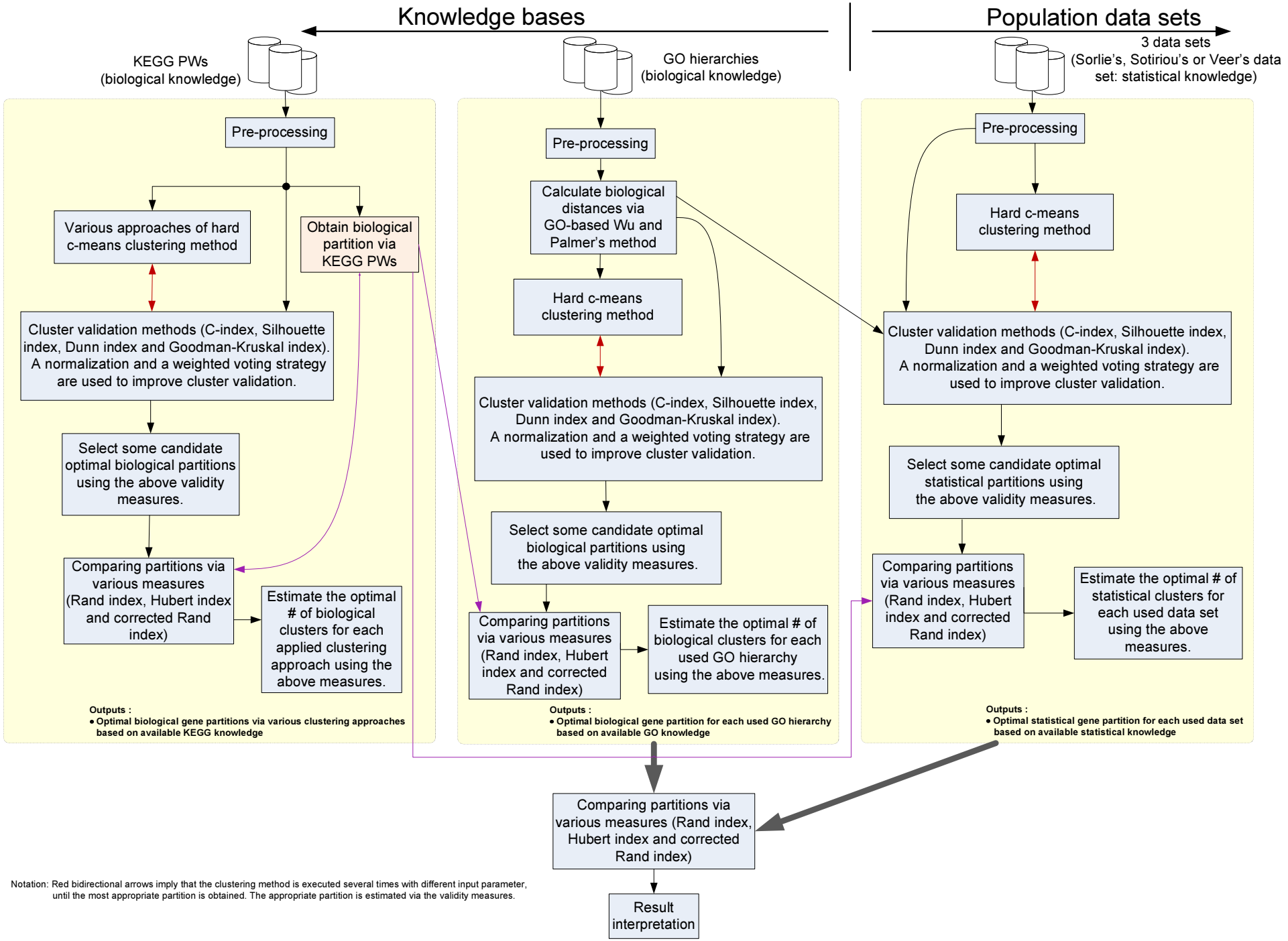


Figure 4 : General structure of the gene clustering methodologies implemented in this thesis.

CHAPTER 2: KNOWLEDGE ORGANIZATION: GENE ONTOLOGY (GO) AND KEGG

2.1 Introduction

To answer meaningful questions, biologists often need to retrieve and analyze data from disparate sources. Biologists currently waste a lot of time and effort in searching all the available information about each small area of research. This is hampered further by the wide variations in terminology, which inhibit effective searching by both computers and people. For example, if someone was searching new targets for antibiotics, he or she might want to find all the gene products[7] that are involved in bacterial protein synthesis and have significantly different sequences or structures from those in humans. If one database uses the phrase "translation" for these molecules, whereas another uses the phrase "protein synthesis", it will be difficult for someone, and even harder for a computer, to find functionally equivalent terms. The Gene Ontology (GO) project is a collaborative effort to address the need for consistent descriptions of gene products in different databases. It provides an ontology of defined terms representing gene

[7] GO uses the term "gene product" to refer collectively to gene and any entities encoded by the gene, e.g. proteins and functional RNAs.

product properties. The ontology covers three domains: cellular component, molecular function and biological process.

In this chapter, we first present the basic concepts and the purpose of the Gene Ontology project. Then, we describe the aforementioned three domains that GO project covers. We also discuss the meaning of GO annotation of genes and proteins, and the tools to accomplish this annotation. Finally, this chapter presents the mappings of concepts from external database systems to equivalent GO terms. Finally, we present another type of available biological knowledge, Kyoto Encyclopedia of Genes and Genomes (KEGG) based on Biological Pathways (PWs).

2.2 GO Project and GO Terms

The GO project is a major bioinformatics initiative with the aim of standardizing the representation of gene and gene product attributes across species and databases. The project provides a controlled vocabulary of terms for describing gene product characteristics and gene product annotation[8] data from GO Consortium[9] members, as well as tools to access and process these data. GO allows us to annotate genes and their products with a limited set of attributes. However, GO does not allow us to describe genes in terms of which cells or tissues they're expressed in, which developmental stages they're expressed at, or their involvement in disease. It is not necessary for GO to do these things because other ontologies are being developed for these purposes. The GO project has developed three structured controlled vocabularies (ontologies) that describe gene products in terms of their associated biological processes, cellular components and molecular functions in a species-independent manner.

The aims of the Gene Ontology project are threefold: first, the development and maintenance of the ontologies themselves, second, the annotation of gene products, which entails making as-

[8] Annotation is the process of assigning GO terms to gene products.

[9] The GO Consortium is the set of biological databases and research groups actively involved in the GO project. This includes a number of model organism databases and multi-species protein databases, software development groups and a dedicated editorial office.

sociations between the ontologies and the genes and gene products in the collaborating databases and third, the development of tools that facilitate the creation, maintenance and use of the ontologies [5]. Each GO term within the ontology has a term name, which may be a word or string of words, a unique alphanumeric identifier, a definition with cited sources and a namespace indicating the domain to which it belongs. Terms may also have synonyms, which are classed as being exactly equivalent or broader or narrower or related to the term name, references to equivalent concepts in other databases and comments on term meaning or usage. An example of a GO term is shown in Figure 5.

```
id:         GO:0000016
name:       lactase activity
namespace:  molecular_function
def:        "Catalysis of the reaction: lactose + H2O = D-glucose + D-galactose." [EC:3.2.1.108]
synonym:    "lactase-phlorizin hydrolase activity" BROAD [EC:3.2.1.108]
synonym:    "lactose galactohydrolase activity" EXACT [EC:3.2.1.108]
xref:       EC:3.2.1.108
xref:       MetaCyc:LACTASE-RXN
xref:       Reactome:20536
is_a:       GO:0004553 ! hydrolase activity, hydrolyzing O-glycosyl compounds
```

Figure 5 : Example of a GO term [6].

The use of GO terms by collaborating databases facilitates uniform queries across the databases. The controlled vocabularies are structured so that they can be queried at different levels. For example, someone can use GO to find all the gene products in the mouse genome that are involved in signal transduction, or can zoom in all the receptor tyrosine kinases. This structure also allows annotators to assign properties to genes or gene products at different levels, depending on the depth of knowledge about them. The GO ontology is structured as a directed acyclic graph: there are no cycles, and “children” can have more than one “parent”, and each term has specific relationships to one or more other terms. The GO vocabulary is designed to be species-neutral, and includes terms applicable to prokaryotes and eukaryotes, as well as to single and multicellular organisms. The GO ontology is not static. Therefore, additions, corrections and alterations are suggested by members of the research and annotation communities, as well as by those directly involved in the GO project. More information about the GO can be found in [59] and [62].

2.3 GO Ontologies

Ontologies provide a vocabulary for representing knowledge and a set of relationships that hold among the terms of the vocabulary. They can be structurally very complex, or relatively simple. Most importantly, ontologies capture domain knowledge in a way that can easily be dealt with by a computer. Because the terms in an ontology and the relationships between the terms are specific, the use of ontologies facilitates to make standard annotations and so the computational queries are improved. As systems make domain knowledge available, to both humans and computers, bio-ontologies such as GO are essential to the process of extracting biological insight from enormous sets of data.

The Gene Ontology is a controlled vocabulary, a set of standard terms, i.e. words or phrases, used for indexing and retrieving information. In addition to defining terms, GO also defines the relationships between the terms, making it a structured vocabulary. The Gene Ontology project provides an ontology of defined terms representing gene product properties. The ontology covers three domains: first, cellular component, i.e. the parts of a cell or its extracellular environment, second, molecular function, i.e. the elemental activities of a gene product at the molecular level, such as binding or catalysis and third, biological process, i.e. operations or sets of molecular events with a defined beginning and end[10] [5]. These operations or sets of molecular events should be pertinent to the functioning of integrated living units: cells, tissues, organs, and organisms. A gene product might be associated with or located in one or more cellular components, it is active in one or more biological processes, during it performs one or more molecular functions. For example, the gene product x can be described by the molecular function term x_1, the biological process terms x_2 and x_3, and the cellular component terms x_4 and x_5. So, the gene product cytochrome c can be described by the molecular function term oxidoreductase activity, the biological process terms oxidative phosphorylation and induction of cell death, and the cellular component terms mitochondrial matrix and mitochondrial inner membrane [5]. These three areas are considered independent of each other. The ontologies are developed to include all terms

[10] Every process should have a discrete beginning and end and these should be clearly stated in the process term definition.

falling into these domains without consideration of whether the biological attribute is restricted to certain taxonomic groups. Therefore, biological processes that occur only in plants (e.g. photosynthesis) or mammals (e.g. lactation) are included. Figure 6 shows a small set of terms from the ontology.

In the diagram in Figure 6, relations between the terms are represented by the colored arrows and the letter in the box midway along each arrow is the relationship type. Note that the terms become more specialized going down the graph, with the most general terms, the root nodes: cellular component, biological process and molecular function, at the top of the graph. Terms may have more than one parent, and they may be connected to parent terms via different relations. As the diagram in Figure 6 suggests, the three GO domains (cellular component, biological process, and molecular function) are each represented by an ontology term. All terms in a domain can trace their parentage to the root term, although there may be numerous different paths via varying numbers of intermediary terms to the ontology root. The three root nodes are unrelated and do not have a common parent node, and hence GO is referred to as three ontologies, or as a single ontology consisting of three sub-ontologies. Some graph-based softwares may require a single root node. In these cases, a “fake” term can be added as a parent of the three existing root nodes, as shown in Figure 7(b).

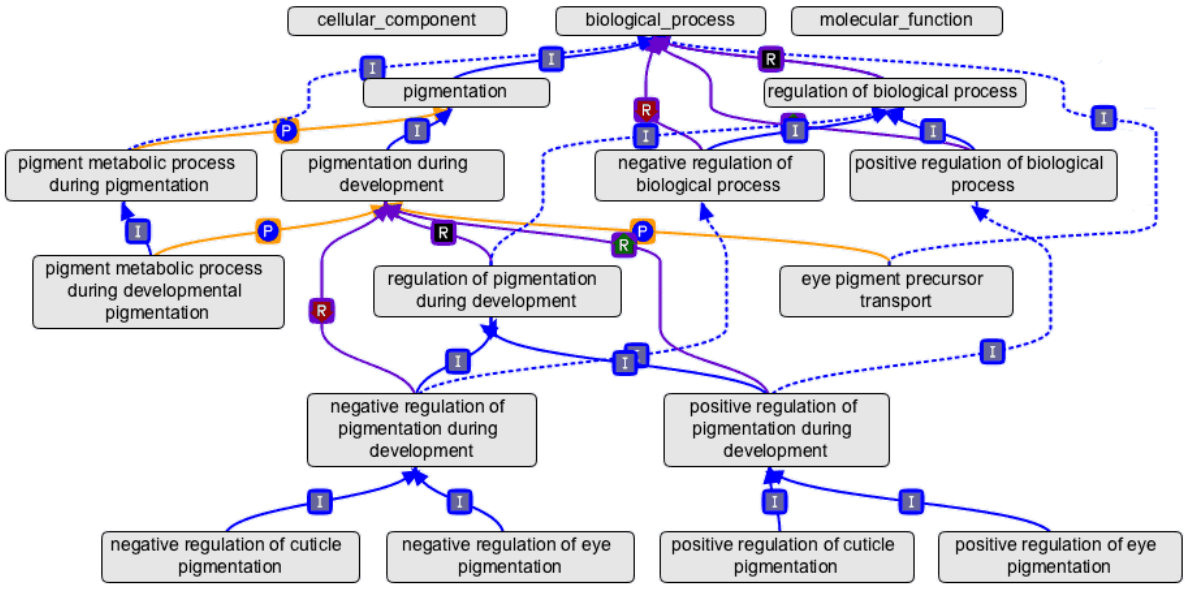


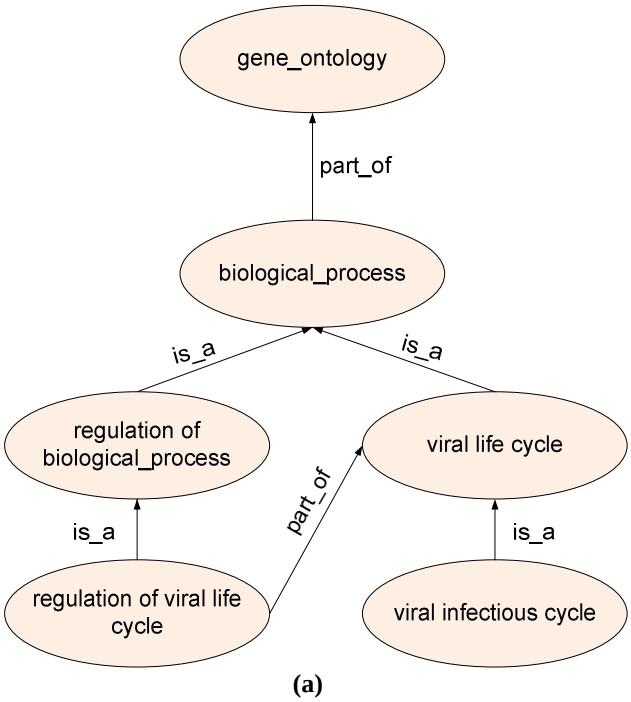
Figure 6 : A set of terms under the biological process node pigmentation [5].

The structure of GO can be described in terms of a directed acyclic graph (DAG), where each GO term is a node, and the relationships between the terms are arcs between the nodes. The relationships used in GO are directed, for example, a mitochondrion is an organelle, but an organelle is not a mitochondrion, and the graph is acyclic. The ontologies resemble a hierarchy, as child terms are more specialized and parent terms are less specialized, but unlike a hierarchy, a term may have more than one parent term. For example, the biological process term hexose biosynthetic process has two parents, hexose metabolic process and monosaccharide biosynthetic process. This is because biosynthetic process is a type of metabolic process and a hexose is a type of monosaccharide. Just as each term is defined, so the relations between GO terms are also categorized and defined.

A hierarchy in the GO may be seen as a network in which each term may represent a "child node" of one or more "parent nodes". There are two types of child-to-parent relationships in the GO: "*is a*" and " *part of* " types. The first type is defined when a child class is a subclass of a parent class. For example, from the BP ontology, "viral infectious cycle" is a child of "viral life cycle". The second type is used when a parent has the child as its part. For instance, from the same ontology, "regulation of viral life cycle" is part of "viral life cycle". Figure 7 illustrates these examples and an another partial view of a DAG in the GO. In more detail now, the "*is a*" relation in GO is very simple: if we say A "*is a*" B, we mean that node A **is a subtype of** node B. For example, mitotic cell cycle "*is a*" cell cycle, or lyase activity "*is a*" catalytic activity. It should be noted that "*is a*" does not mean "*is an instance of* ". An "*instance*", ontologically speaking, is a specific example of something, e.g. a cat "*is a*" mammal, but Garfield **is an instance of** a cat, rather than a subtype of cat. The "*is a*" relation is transitive, which means that if A "*is a*" B, and B "*is a*" C, we can infer that A "*is a*" C. An example is shown in Figure 8. So, from this example we can see that mitochondrion "*is an*" intracellular organelle and intracellular organelle "*is an*" organelle, therefore mitochondrion "*is an*" organelle.

The relation " *part of* " is used to represent part-whole relationships in the Gene Ontology. This relation has a specific meaning in GO and it would be added between A and B, only if B is necessarily " *part of* " A. That means wherever B exists, it is a part of A, and the presence of B implies the presence of A. However, given the occurrence of A, we cannot say that B exists, i.e. **all** B are "*part of* " A, but **some** A "*have part*" B. An example is shown in Figure 9 which

presents that replication fork is necessarily " *part of* " chromosome: **all** replication are " *part of* " some chromosomes, but only **some** chromosomes "*have part*" replication fork. Like the "*is a*" relation, the " *part of* " relation is transitive too, as Figure 10 shows : mitochondrion is " *part of* " cytoplasm and cytoplasm is " *part of* " cell, therefore mitochondrion is " *part of* " cell. Also, if a " *part of* " relation is followed by an "*is a*" relation, it is equivalent to a " *part of* " relation, i.e. if A is " *part of* " B, and B "*is a*" C, we can infer that A is " *part of* " C. In Figure 11 we see that mitochondrial membrane is " *part of* " mitochondrion, and mitochondrion "*is an*" intracellular organelle, therefore mitochondrial membrane is " *part of* " intracellular organelle. It should be noted that if the order of the relationships is reversed, the result is the same, i.e. mitochondrion "*is a*" intracellular organelle and intracellular organelle is " *part of* " cell, therefore mitochondrion is " *part of* " cell. The logical rules regarding the " *part of* " and "*is a*" relations hold no matter how many intervening "*is a*" and " *part of* " relations there are. In Figure 12, the nodes between mitochondrion and cell are connected by both "*is a*" and " *part of* " relations, however this is equivalent to saying mitochondrion is " *part of* " cell.



(a)

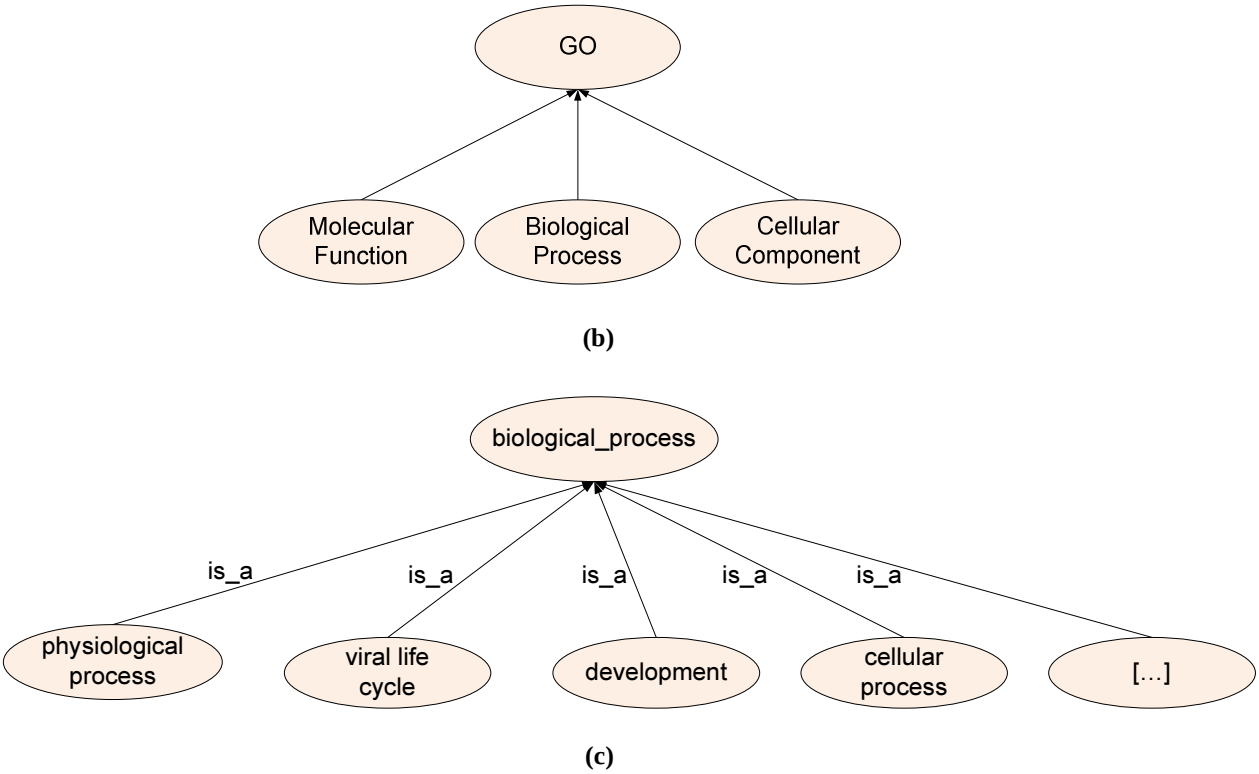


Figure 7 : Different views of the GO: **(a)** Example of a DAG. **(b)** GO taxonomies. **(c)** Partial view of the first level of BP. […] indicates the presence of several terms not included here.

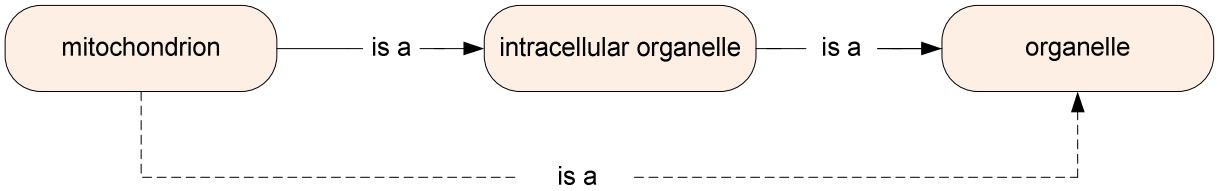


Figure 8 : Transitivity of the "is a" relation.

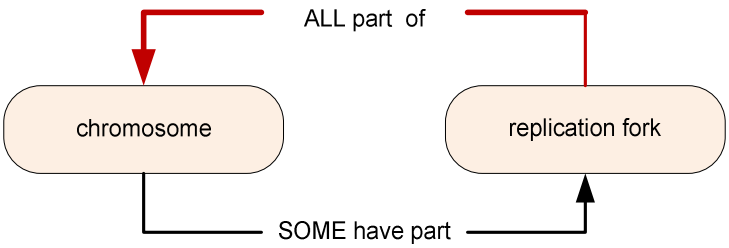


Figure 9 : An example of the " *part of* " relation.

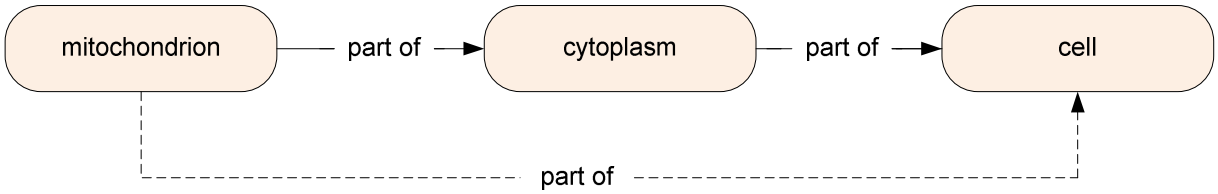


Figure 10 : Transitivity of the " *part of* " relation.

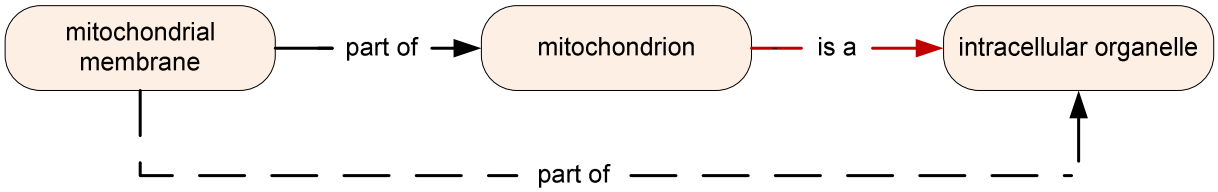


Figure 11 : An example that shows a " *part of* " relation to be followed by an "*is a*" relation.

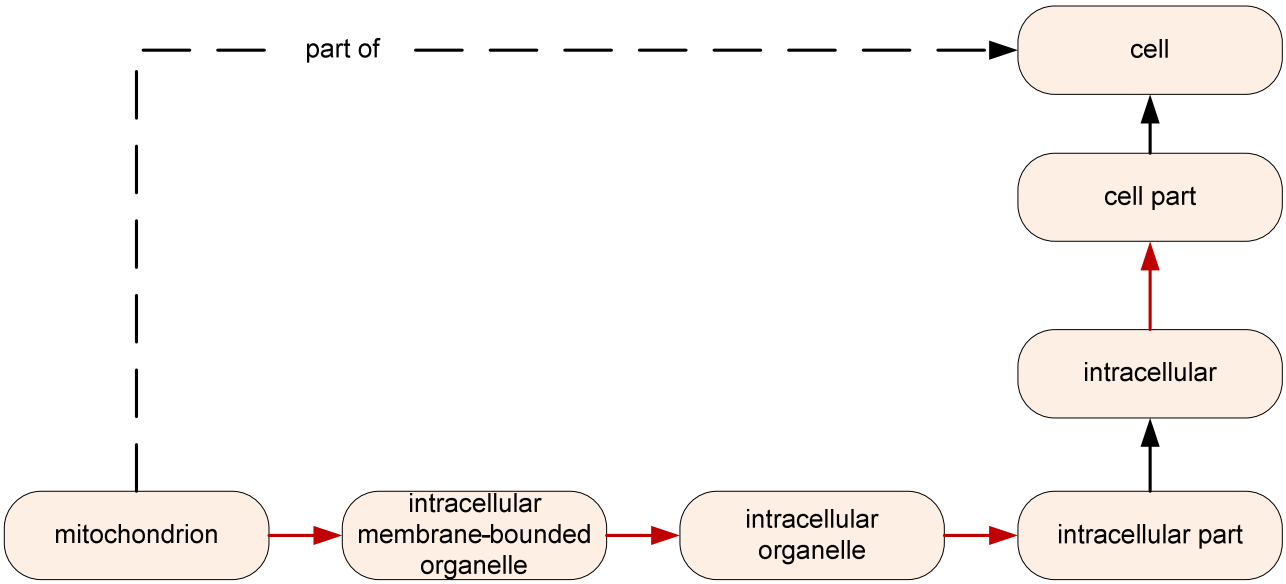


Figure 12 : An example of both "*is a*" and " *part of* " relations.

Another common relationship in the GO is that where one process directly affects the manifestation of another process or quality, i.e. the former "*regulates*" the latter. The target of the regulation may be another process, for example, regulation of a pathway or an enzymatic reaction, or it may be a quality, such as cell size or pH. Analogously to " *part of* ", this relation is used specifically to mean necessarily "*regulates*". That means whenever B is present, it **always** "*regulates*" A, but A may **not always** be "*regulated by*" B. In Figure 13 we see an example of this relationship. Whenever a cell cycle checkpoint occurs, it always "*regulates*" the cell cycle. However, the cell cycle is not solely "*regulated by*" cell cycle checkpoints, as there are also other

processes that regulate it. The regulation of a process does not need to be part of the process itself. That means that regulation of transcription describes the processes that modulate the activity of the transcriptional machinery, but those processes are not an integral part of transcription.

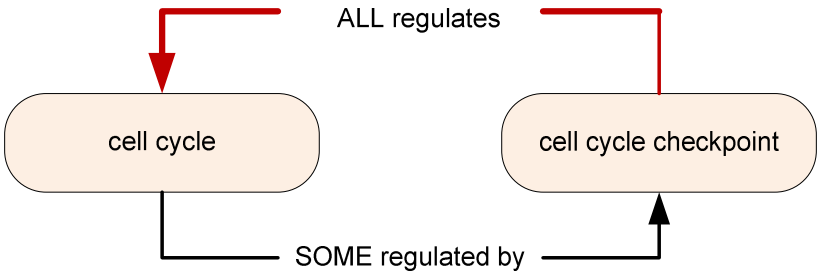


Figure 13 : An example of the "*regulates*" relation.

Cellular Component (CC)

Cellular component refers to the unique, highly organized substances of which cells, and so living organisms, are composed. Examples include membranes, organelles, proteins, and nucleic acids. Whilst the majority of cellular components are located within the cell itself, some may exist in extracellular areas of an organism. The cellular component ontology describes locations, at the levels of subcellular structures and macromolecular complexes. Examples of cellular components include nuclear inner membrane, with the synonym inner envelope, and the ubiquitin ligase complex, with several subtypes of macromolecular complexes. Generally, a gene product is located in or is a subcomponent of a particular cellular component. The cellular component ontology includes multi-subunit enzymes and other protein complexes, but not individual proteins or nucleic acids. Cellular component also does not include multicellular anatomical terms. The cellular component ontology is an "*is a*" complete tree, meaning that every term has a path to the root node which passes solely through "*is a*" relationships.

Molecular Function (MF)

Molecular function covers the elemental activities of a gene product at the molecular level, such as binding or catalysis. GO molecular function terms represent activities rather than the entities (molecules or complexes) that perform the actions, and do not specify where or when or in what context, the action takes place. Molecular functions generally correspond to activities that can be performed by individual gene products, but some activities are performed by assembled complexes of gene products. Examples of broad functional terms are catalytic activity, transporter activity or binding. Examples of narrower functional terms are adenylate cyclase activity or

Toll receptor binding. It is easy to confuse a gene product name with its molecular function, and for that reason many GO molecular functions are appended with the word “activity”.

Biological Process (BP)

A biological process is a recognized series of events or molecular functions. In other words, a biological process is a process of a living organism. A process is a collection of molecular events with a defined beginning and end, as it has been already mentioned. Biological processes are regulated by many means. Some examples include the control of gene expression, protein modification or interaction with a protein or substrate molecule. Examples of broad biological process terms are cellular physiological process or signal transduction. Examples of more specific terms are pyrimidine metabolic process or alpha-glucoside transport. It can be difficult to distinguish between a biological process and a molecular function, but the general rule is that a process must have more than one distinct steps. A biological process is not equivalent to a pathway. At present, GO does not try to represent the dynamics or dependencies that would be required to describe fully a pathway. The biological process ontology includes terms that represent collections of processes, as well as terms that represent a specific, entire process. Generally, the former will have mainly “*is a*” children, and the latter will have “ *part of* ” children that represent subprocesses. An example of such relationships is shown in Figure 14.

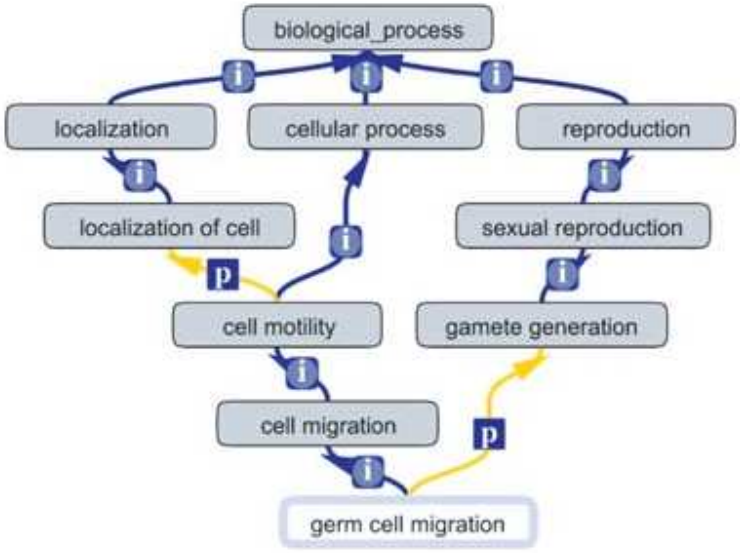


Figure 14 : The GO vocabularies are sets of defined terms and specifications of the relationships between them. As indicated in this diagram, the GO vocabularies are directed acyclic graphs. In this example, germ cell migration has two parents, it is a “ part of ” gamete generation and “is a” (is a subtype of) cell migration. The GO uses these elementary relationships in all vocabularies.

It is important to note that the functions of a gene product are the jobs that it does or the "abilities" that it has. These may include transporting things around, binding to things, holding things together and changing one thing into another. This is different from the biological processes the gene product is involved in, which involve more than one activity. One way to understand this is to consider the analogy of a company or organization [5]. Individuals (gene products) have different abilities or tasks (functions) and they work together to achieve different goals (processes). It is easy to confuse a job title (gene product name) with a function. For example, "secretarial activity" may seem like a valid function because we have a good conceptual idea of what a secretary does. However, in different companies, secretaries might do different things. One secretary might have the functions "typing", "answering phone" and "making coffee", whilst another might have these functions and additionally "photocopying". In the Gene Ontology, a function should be unambiguous and it should mean the same thing no matter what species we are dealing with.

2.4 GO Annotation and Tools

Annotation is the practice of capturing the activities and localization of a gene product with GO terms, providing references and indicating what kind of evidence is available to support the annotations. In other words, annotation is the process of assigning GO terms to gene products. Because a single gene may encode different products with very different attributes, GO recommends associating GO terms with database objects representing gene products rather than genes. If identifiers are not available to distinguish individual gene products, GO terms may be associated with an identifier for gene and thus, gene is associated with all GO terms applicable to any of its products. In addition to the gene product identifier and the relevant GO term, GO annotations have the following data: first, every annotation must be attributed to a source, the reference used to make the annotation (e.g. a literature reference, another database or a computational analysis), second, the annotation must indicate what kind of evidence is found in the cited source to support the association between the gene product and the GO term (i.e. an evidence code[11] de-

[11] A simple controlled vocabulary is used to record evidence. The evidence codes are simply the three-letter codes used to signify the type of evidence cited. The evidence codes come from the Evidence Code Ontology.

noting the type of evidence upon which the annotation is based) and finally, the date and the creator of the annotation. Full annotation data sets can be downloaded from the GO website. A gene product can be annotated to zero or more nodes of each ontology, at any level within each ontology. Also, annotation of a gene product to one ontology is independent of its annotation to other ontologies. Annotations should reflect the normal function, process, or localization (component) of the gene product. An example of a GO annotation is shown in Figure 15. GO is a work in progress, so not all genes and proteins have GO terms associated with them yet.

```
Gene product:     Actin, alpha cardiac muscle 1, UniProtKB:P68032
GO term:          heart contraction ; GO:0060047 (biological process)
Evidence code:    Inferred from Mutant Phenotype (IMP)
Reference:        PMID:17611253
Assigned by:      UniProtKB, June 06, 2008
```

Figure 15 : Example of a GO annotation [6].

In addition, the GO consortium has prepared GO slims, which are "slimmed down" versions of the ontologies that allow someone to assign GO slims terms to genomes or sets of gene products and thus to gain a high-level view of gene functions. GO slims are cut-down versions of the GO ontologies containing a subset of the terms in the whole GO. They give a broad overview of the ontology content without the details of the specific fine grained terms. GO slims are created by users according to their needs and may be specific to species or to particular areas of the ontologies. GO slims are particularly useful for giving a summary of the results of GO annotation of a genome, microarray or cDNA collection when broad classification of gene product function is required. Using GO slims someone can, for example, work out what proportion of a genome is involved in signal transduction, biosynthesis or reproduction.

There is a large number of tools available both online and to download that use the data provided by the GO project. The vast majority of these come from third parties, while the GO Consortium develops and supports two tools, AmiGO [60] and OBO-Edit [61]. Members of the GO Consortium make their annotation data freely available to the public as part of the data accessed by AmiGO, the GO browser and search engine. AmiGO provides an interface to search and browse the ontology and annotation data provided by the GO consortium. Users can search for gene products and view the terms with which they are associated. Alternatively, users can search or browse the ontology for GO terms of interest and see term details and gene product annotations. AmiGO also provides a BLAST search engine, which searches the sequences of genes and

gene products that have been annotated to a GO term and submitted to the GO Consortium. AmiGO accesses the GO mySQL database. Annotation data sets from individual databases can be found on the GO annotations page.

OBO-Edit is an open source, platform-independent ontology editor developed and maintained by the Gene Ontology Consortium. It is implemented in Java, and uses a graph-oriented approach to display and edit ontologies. Its emphasis on the overall graph structure of an ontology provides a friendly interface for biologists and makes OBO-Edit excellent for the rapid generation of large ontologies. OBO-Edit includes a comprehensive search and filter interface, with the option to render subsets of terms to make them visually distinct. The user interface can also be customized according to user preferences. OBO-Edit has also a reasoner that can infer links that have not been explicitly stated, based on existing relationships and their properties. Although it was developed for biomedical ontologies, OBO-Edit can be used to view, search and edit any ontology. It is freely available to download.

2.5 Mappings of External Databases to GO

The Gene Ontology is not the only attempt to build structured controlled vocabularies for genome annotation. Thus, to aid users, the GO Consortium provides mappings of its terms to terms in a number of external vocabularies. Each vocabulary has its own nomenclature, for example GenBank Accession, Clone Id, Unigene Cluster, EntrezGene are some of the existing nomenclatures. Mappings are files that contain classes or entities from external classification systems, such as Enzyme Commission numbers, UniProt keywords or ProSite domains, indexed to identical or similar or related GO terms. Although the GO Consortium endeavours to make mappings as accurate as possible, it cannot guarantee that the mappings provided by the GO project are either complete or exact. This may be due to the absence of definitions from GO terms or from terms in some external systems. Furthermore, the GO ontologies and the external databases may have changed since the mappings were made. It is also noted that mapping of any existing vocabulary to any existing vocabulary is feasible via a variety of freely available tools, such as Clone/Gene ID Converter and SOURCE Batch Search tools.

2.6 Biological Pathways (PWs)

A biological pathway is a series of actions among molecules in a cell that leads to a certain product or a change in a cell. Such a pathway can trigger the assembly of new molecules, such as a fat or protein. Pathways can also turn genes on and off, or spur a cell to move. For one's body to develop properly and stay healthy, many things must work together at many different levels, from organs to cells. Cells are constantly receiving cues from both inside and outside the body, which are prompted by such things as injury, infection, stress or even food. To react and adjust to these cues, cells send and receive signals through biological pathways. The molecules that make up biological pathways interact with signals, as well as with each other, to carry out their designated tasks. Biological pathways can act over short or long distances. For example, some cells send out signals to nearby cells to repair localized damage, such as a scratch on one's knee. Other cells produce substances, such as hormones, that travel through one's blood to distant target cells. Biological pathways can also produce small or large outcomes. For example, some pathways subtly affect how the body processes drugs, while others play a major role in how a fertilized egg develops into a baby. There are many other examples of how biological pathways help one's body to work. For example, the pupil in one's eye opens or closes in response to light, or if one's skin senses that the temperature is rising, the body sweats to cool him or her down. In fact, without biological pathways, we and all other living creatures could not exist. Still, it's important to keep in mind that biological pathways do not always work properly. When something goes wrong in a pathway, the result can be a disease such as cancer or diabetes.

There are many types of biological pathways. Some of the most common are involved in metabolism, the regulation of genes and the transmission of signals [48]. Metabolic pathways make possible the chemical reactions that occur in our bodies. An example of a metabolic pathway is the process by which one's cells break down food into energy molecules that can be stored for later use. Other metabolic pathways actually help to build molecules. Gene regulation pathways turn genes on and off. Such action is vital because genes produce proteins, which are the key components needed to carry out nearly every task in our bodies. Proteins make up our muscles and organs, help our bodies move and defend us against germs. Signal transduction pathways move a signal from a cell's exterior to its interior. Different cells are able to receive specific signals through structures on their surface, called receptors. After interacting with a receptor, the

signal travels through the cell where its message is transmitted by specialized proteins that trigger a specific action in the cell. For example, a chemical signal from outside the cell might be turned into a protein signal inside the cell. In turn, that protein signal may be converted into a signal that prompts the cell to move. Figure 16 shows an example of a biological pathway, which is the biological pathway for Huntington's Disease. This pathway governs the movement of information between genes and proteins, processes and locations in the cell. This one is a relatively simple pathway. More complex pathways can have hundreds of elements each directional.

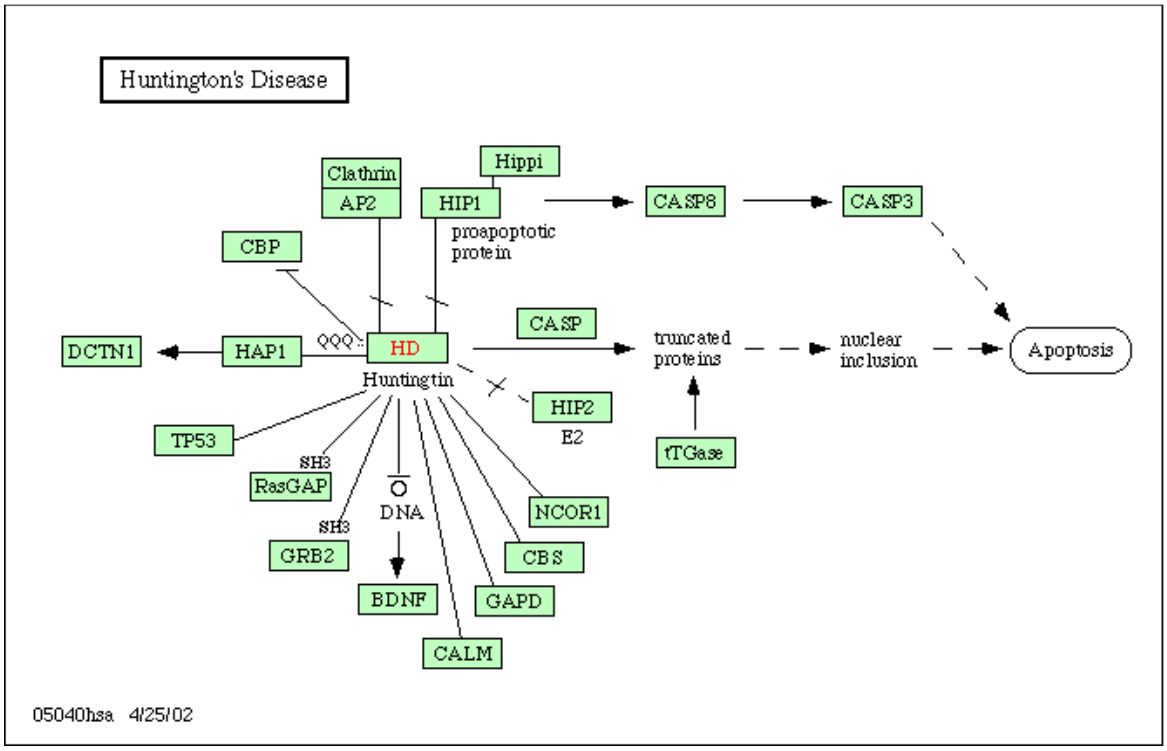


Figure 16 : The biological pathway for Huntington's Disease.

Researchers are learning that biological pathways are far more complicated than once thought. Most pathways do not start at point A and end at point B. In fact, many pathways have no real boundaries, and they often work together to accomplish tasks. When multiple biological pathways interact with each other, it is called a biological network. An example of a biological network is presented in Figure 17.

Many important biological pathways have been discovered through laboratory studies of cultured cells, bacteria, fruit flies, mice and other organisms. Many of the pathways identified in these model systems are the same or have similar counterparts in humans. Still, many biological

pathways remain to be found. It will take years of research to identify and understand the complex connections among all of the molecules in all biological pathways, as well as to understand how these pathways work together.

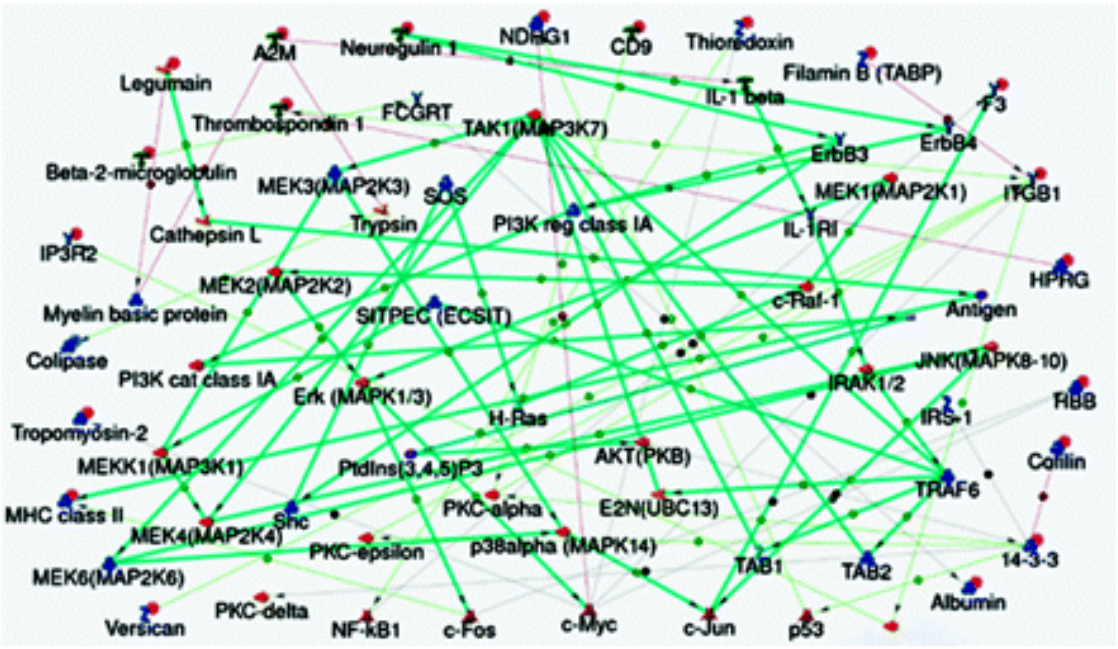


Figure 17 : Biological network analysis of differentially expressed proteins in both pancreatic cancer and chronic pancreatitis.

Researchers are also able to learn a lot about human disease from studying biological pathways. Identifying what genes, proteins and other molecules are involved in a biological pathway can provide clues about what goes wrong when a disease strikes. For example, researchers may compare certain biological pathways in a healthy person to the same pathways in a person with a disease to discover the roots of the disorder. Keep in mind that problems in any number of steps along a biological pathway can often lead to the same disease. Finding out what pathway is involved in a disease and identifying which step of the pathway is affected in each patient may lead to more personalized strategies for diagnosing, treating and preventing disease. Researchers currently are using information about biological pathways to develop new and better drugs. It likely will take some time before we routinely see drugs that are specifically designed using the pathway approach. However, doctors already use pathway information to choose and combine existing drugs more effectively.

For example, take the case of cancer [48]. Until recently, many had hoped that most types of cancers were driven by a single genetic error and could be treated by designing drugs to target those specific errors. Much of that hope was based on the success of imatinib (Gleevec), a drug that was specifically designed to treat a blood cancer called chronic myeloid leukemia (CML). CML occurs because of a single genetic glitch that leads to the production of a defective protein that spurs uncontrolled cell growth. Gleevec binds to that protein, stopping its activity and producing dramatic results in many CML patients. Unfortunately, the one-target, one-drug approach has not held up for most other types of cancer. Recent projects that deciphered the genomes of cancer cells have found an array of different genetic mutations that can lead to the same cancer in different patients. Then, based on the genetic profile of their particular tumor, patients could receive the drug or drug combination that is most likely to work for them. The complexity of the findings appears daunting. Instead of attempting to discover ways to attack one well-defined genetic enemy, researchers now faced the prospect of fighting lots of little enemies. Fortunately, this complex view can be simplified by looking at biological pathways that are disrupted by the genetic mutations. Rather than designing dozens of drugs to target dozens of mutations, drug developers could focus their attentions on just two or three biological pathways. Patients could then receive the one or two drugs most likely to work for them based on the pathways affected in their particular tumors. One way to understand this is to imagine a thousand people travelling towards the front door of a single building. In order to keep all these people from entering the building there are two ways. If you had limitless resources, you could hire workers to go out and stop each person. That would be the one-target, one-drug approach. But if you wanted to save a lot of time and money, you could just block the door to the building. That is the pathway-based strategy that many researchers are now pursuing to design drugs for cancer and other common diseases.

2.7 Kyoto Encyclopedia of Genes and Genomes (KEGG)

Kyoto Encyclopedia of Genes and Genomes (KEGG) is a collection of 16 online databases dealing with genomes, enzymatic pathways, and biological chemicals. KEGG connects known information on molecular interaction networks, such as pathways and complexes (this is the Pathway Database), information about genes and proteins generated by genome projects (this is

the Gene Database) and information about biochemical compounds and reactions (these are the Compound and the Reaction Databases). These last databases are different networks, known as the protein network and the chemical universe respectively.

KEGG is widely used in biology, biochemistry and medicine to study metabolic and regulatory processes. The presentation of these processes as pathway diagrams greatly helps researchers in understanding key functions of biological systems. The developers consider KEGG to be a "computer representation" of the biological systems. The pathway data can be studied in a visual way and is also available as KEGG Markup Language (KGML) files. Thus it can be used as a basis for simulation models. KEGG has been widely used as a reference knowledge base for biological interpretation of large-scale datasets generated by sequencing and other high-throughput experimental technologies [10]. However, the graphical presentation of pathway information in KEGG is restricted to semi-static visualization and editing KGML files is not simple.

KEGG pathway is a collection of manually drawn pathway maps representing our knowledge on the molecular interaction and reaction networks for Global Map, Metabolism, Genetic Information Processing, Environmental Information Processing, Cellular Processes, Organismal Systems, Human Diseases and also on the structure relationships in Drug Development (KEGG drug structure maps). KEGG Atlas is an advanced graphical interface to explore the KEGG pathway maps with zooming and navigation capabilities [10].

2.8 Summary

The aim of GO project is to standardize the representation of gene and gene product attributes across species and databases. The GO project has developed three structured ontologies: biological process, cellular component and molecular function. The existence of the ontologies is to provide domain knowledge that can be easily processed by a computer. The aims of the GO project are threefold: the development and maintenance of the ontologies themselves, the annotation of gene products and the development of tools that facilitate the creation, maintenance and use of the ontologies. Annotation is the practice of capturing the activities and localization of a gene product with GO terms, providing references and indicating what kind of evidence is available to support the annotations. The structure of GO can be described in terms of a directed acyc-

lic graph (DAG), where each GO term is a node, and the relationships between the terms are directed arcs between the nodes. A hierarchy in the GO may be seen as a network in which each term may represent a "child node" of one or more "parent nodes". The two types of child-to-parent relationships are the "*is a*" and " *part of* " types . Another common relationship is the "*regulates*" type. There is a variety of tools available that use the data provided by the GO project. Also, the GO Consortium provides mappings of its terms to a number of external vocabularies. Except the GO, another type of available biological knowledge is KEGG. It is noted that in this study, we take advantage of both GO and KEGG knowledge.

CHAPTER 3: GENE CLUSTERING BASED ON STATISTICAL AND BIOLOGICAL KNOWLEDGE

3.1 Introduction

Several clustering algorithms have been suggested to analyze genome expression data, but fewer solutions have been implemented to guide the design of clustering-based experiments and assess the quality of their outcomes. Clustering can support the identification of existing underlying relationships among a set of variables such as biological conditions or perturbations [13]. It may represent a basic tool not only for the classification of known categories, but also for the discovery of relevant classes. In genome expression domain it has provided the basis for novel

clinical diagnostic and prognostic studies. One major data analysis step is to integrate the numerical analysis, which is derived from the implementation of clustering algorithms of co-expressed genes, with biological function information. Many approaches and tools have been proposed to address this problem at different processing levels. Some methods, for example, score whole clustering outcomes or specific clusters according to their biological relevance, while other techniques aim to estimate the significance of overrepresented functional annotations, such as those encoded in the Gene Ontology (GO), in clusters. Also, some other approaches directly incorporate biological knowledge into the clustering process to aid in the detection of relevant clusters of co-expressed genes involved in common processes. Several tools have been developed for ontological analysis of gene expression data and more tools are likely to be proposed in the future.

Clustering techniques are designed to uncover existing groups in data, usually with very limited information available. For example, not only the membership of the data points has to be determined, but often also the number of groups. The main objective of the research is an application of the clustering and cluster validity methods to estimate the number of clusters in datasets. The prediction of the correct number of clusters in a data set is a fundamental problem in unsupervised learning. Various cluster validity indices have been proposed to measure the quality of clustering results [8], [9]. Recent studies confirm that there is no universal pattern recognition and clustering model to predict molecular profiles across different datasets. Thus, it is useful not to rely on one single clustering or validation method, but to apply a variety of approaches. Therefore, combination of GO-based (knowledge-driven) and microarray data (data-driven) validation methods may be used for the estimation of the number of clusters. This estimation approach may perform an useful tool to support biological and biomedical knowledge discovery. A normalization and a weighted voting technique are usually used to improve the prediction of the number of clusters based on different data mining techniques. More details about these techniques will be discussed in Sections 3.7 and 3.8 respectively.

The many available procedures are based on various optimality criteria and since different criteria can be used, it is important to be able to compare results obtained by different approaches. Similarly, one may be interested in assessing degree of similarity (or verifying equivalence) of two clustering algorithms (for example one being a simpler and/or more efficient version of the other). This is an important issue with current research, where large data sets are so common.

The problem of comparing two different partitions of a finite set of objects reappears continually in the clustering literature. So, a variety of similarity indices were designed to compare partitions (clusterings) of a data set. Furthermore, the behavior of the similarity index can also be used as an indicator of the proper number of clusters in a data set.

In this chapter, we present all the above concepts in detail. We present the implemented data- and knowledge-driven clustering approaches and cluster validity assessment system. Normalization and weighted voting techniques are used to improve the prediction of the correct number of clusters. Also, we discuss about several measures, some of which have been implemented, to compare different partitions obtained from different clustering approaches.

3.2 Genomic Expression Data and Biological Knowledge Databases

The DNA microarray technologies allow to compare the expression of thousands of genes in different tissues, cells or physiological conditions. It can be used for diagnosis, therapy, follow-up of a treatment or even for characterizing physiological states. Indeed, the major interest of these technologies is to identify, among multiple candidate genes, which ones are the most likely to be involved in a considered trait. So, online biological knowledge databases (KEGG, GO, RIKEN), biological repositories for gene expression array-based data (GEO) and bibliographical database (PubMed) have recently been developed. However, the size and heterogeneity of such databases remain problematic.

In this thesis, in order to take advantage of the available biological information an enrichment cluster analysis using GO terms or KEGG pathways is carried out. The clustering approaches implemented in this thesis, carry out both analyses. The Gene Ontology (GO) and Kyoto Encyclopedia of Genes and Genomes (KEGG) databases try to arrange genes to specific informative groups. The GO database is divided into three different ontologies called Molecular Function, Cellular Component and Biological Process. The structure of the database is an acyclic directed graph. To each node (GO term) a set of genes is annotated. The root is the most unspecific GO term. Its set of genes consists of every gene in the database. The leaves are the most specific GO terms. An illustrating example is the GO term GO:0003713 which stands for transcription co activator activity, containing 392 genes. One child of that term is GO:0008140 (cAMP response

element binding protein binding). Indeed, this is a more specific term as the first one, containing only 7 genes. The KEGG database provides a sorting of genes, depending to which biological pathway they belong. Each KEGG identifier stands for one and has the set of genes from the pathway annotated. With the help of the above curated vocabularies the gene lists resulting from the experiments can be further analyzed, without being an expert in all fields of molecular biology. To do this someone can ask if a specific GO term or KEGG pathway is overrepresented in the gene lists. The resulting terms and pathways from the analysis could be used to describe the set of differential expressed genes or the found cluster in a meaningful biological way. So, the choice of these two types of biological knowledge (i.e. GO and KEGG) is not arbitrary, since in some sense GO and KEGG give similar biological knowledge. In this way, the results obtained using the available GO knowledge can be compared with the results obtained using the available KEGG knowledge, in order to make meaningful biological conclusions.

It is noted that the genes, which we have at our disposal, are in the Unigene nomenclature. As far as biological knowledge is concerned, we annotate the available genes to the GO in order to take the available biological knowledge from the GO. This step incorporates also the differentiation of the GO terms that refer to the BP hierarchy from those GO terms that refer to the MF hierarchy. So, we lead to the biological knowledge that refers to the BP hierarchy and the biological knowledge that refers to the MF hierarchy respectively. It is noted, that we do not use the CC hierarchy at all, as it has " *part of* " relations. Only "*is a*" relations are allowed in our study, which are present to the BP and MF hierarchies. The reason for this restriction is discussed in Section 3.3. Furthermore, we map the available genes to the Entrez Gene nomenclature so as to take advantage of the available biological information from KEGG, where genes are named in the Entrez Gene nomenclature. It is noted that the annotation to the GO and the mapping to the Entrez Gene nomenclature are done via the Clone/Gene ID Converter[12] and the SOURCE Batch Search[13] tools respectively.

[12] It is freely available at http://idconverter.bioinfo.cnio.es/IDconverter.php.

[13] It is freely available at http://smd.stanford.edu/cgi-bin/source/sourceBatchSearch.

Finally, as far as genomic expression data sets are concerned, this research is based on three data sets. The first data set, the Sorlie's data set, comprises 6832 genes with 59 patients' samples per gene. The second one, the Veer's data set, includes 79 patients' samples described by the expression levels of 14639 genes, while the last data set, the Sotiriou's data set, has 2941 genes with 99 patients' samples each one. All genes are in the Unigene nomenclature, as it has been mentioned before. Before these data sets are ready to be used, we impute the NA (Not Available) values and then we select only the common genes among the three data sets. There are various methods to impute the NA values. One method is just to ignore the genes that have such values, while another method is to take the mean value of the gene, ignoring at the moment the NA values and as a second step replace all these NA entries with this mean value. The mean value in turn is computed taking into account all patients' samples concerning that gene. A third method is to use the k-nearest neighbor (embedded function in matlab) in order to fill these NA values. The second method is chosen, since it is not recommended to reduce the number of genes, as caused by the first method, as well as it is not recommended to impute the NA values of a gene using its neighbor genes' values, as the third method does.

It is also important to note that in order to compare the results obtained using the available GO knowledge, the available KEGG knowledge or the available three data sets, we finally select only the common genes among the three data sets that can be annotated to the GO and can be also mapped to the Entrez Gene nomenclature. After the above pre-processing, we finally keep 946 distinct genes, that means that we keep 32% of the genes from Sotiriou's data set, 14% of the genes from Sorlie's data set and 6% of the genes from Veer's data set. The above results are acceptable, since we finally keep the most important genes concerning breast cancer.

3.3 GO-Based Similarity Measurement Techniques

The automated integration of background knowledge is fundamental to support the generation and validation of hypotheses about the functionality of gene products. One such source of prior knowledge is the Gene Ontology (GO). We will present an approach for gene clustering and assessing cluster validity based on similarity knowledge extracted from the Gene Ontology (GO) and databases annotated to the GO. One of the main objectives of this research is to use knowledge-driven gene clustering approaches and knowledge-driven cluster validity methods to

estimate the number of clusters in a data set. Thus, a knowledge-driven cluster validity assessment system for microarray data is implemented. More details can be found in Sections 3.4 and 3.6. Different methods exist to measure similarity between genes products based on the GO. The method implemented in this study, processes overall similarity values, which are calculated by taking into account the combined annotations originating from the three GO hierarchies [21].

A traditional node-counting method has been implemented to measure knowledge-based similarity between genes products (biological distances). Unlike traditional methods that only use (gene expression) data-derived indices, this method consists of validity indices that incorporate similarity knowledge originating from the GO and a GO-driven annotation database. A traditional edge-counting method proposed by Wu and Palmer [20] is implemented to measure similarity between genes products. Edge-counting approach calculates the distance between the nodes associated with these terms in a hierarchy. Given a pair of terms, c_1 and c_2, this traditional method for measuring their similarity, consists of calculating the distance measured by the number of edges between the nodes associated with these terms in the ontology. The shorter this distance is, the higher the similarity is. The shortest or the average distance may be used when there are multiple paths. This type of approaches is commonly known as edge-counting methods. Variations may define weights for the links according to their position in the taxonomy. One of the main limitations shown by these methods is that they assume that nodes and links are uniformly distributed in an ontology, e.g. in the GO. This is not an accurate assumption in taxonomies exhibiting variable link densities. Information-theoretic models [23] offer alternative approaches to measuring similarity in an ontology. Previous research has shown that this type of approaches may be significantly less sensitive to link density variability [22], [24]. These methods traditionally consider only the "*is a*" links in a taxonomy. However, it has been shown that other types of links may also be processed to perform similarity assessment [22]. The majority of the GO links are "*is a*" links [25].

Topological and statistical information extracted from the GO and databases annotated to the GO may be used to measure similarity between gene products. Different GO-driven similarity assessment methods may be then implemented to perform clustering or to quantify the quality of the resulting clusters. Cluster validity assessment may consist of data- and knowledge-driven methods, which aim to estimate the optimal cluster partition from a collection of candidate parti-

tions. Data-driven methods mainly include statistical tests or validity indices applied to the data clustered.

For a given pair of gene products, g_1 and g_2, sets of GO terms $T_1 = t_i$ and $T_2 = t_j$ are used to annotate these genes. Before estimating *between-gene* similarity it is first necessary to understand how to measure *between-term* similarity. Similarity was defined by Wu and Palmer [20] as follows:

$$sim\left(t_i, t_j\right) = \begin{cases} \dfrac{2 \cdot N}{N_i + N_j + 2 \cdot N} & \text{if } N \neq 0 \\ \dfrac{2}{N_i + N_j} & \text{if } N = 0 \end{cases}, \tag{3.1}$$

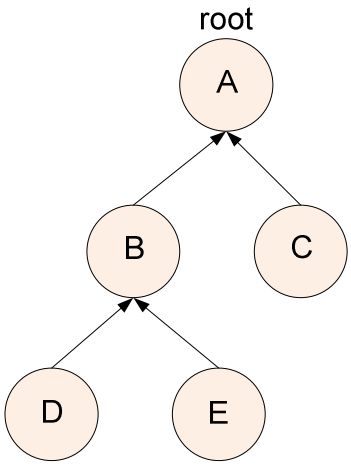
where N_i and N_j are the **minimum** number of links (edges) from t_i and t_j to their closest common parent in the GO hierarchy, T_{ij}, and N is the **maximum** number of links from T_{ij} to the GO hierarchy root. It is noted that when $N \neq 0$, it holds that

$sim\left(t_i, t_j\right) = \dfrac{2 \cdot N}{N_i + N_j + 2 \cdot N} = \dfrac{N}{(N_i + N_j + 2 \cdot N)/2} = \dfrac{N}{\frac{1}{2} \cdot \left(N_i + N_j\right) + N}$ and so, when $N = 0$, it holds that $sim\left(t_i, t_j\right) = \dfrac{1}{\frac{1}{2} \cdot \left(N_i + N_j\right)} = \dfrac{2}{\left(N_i + N_j\right)}$. Thus, we conclude to equation (3.1). This similarity assessment metric may be transformed into a distance, d, metric:

$$d\left(t_i, t_j\right) = 1 - sim\left(t_i, t_j\right). \tag{3.2}$$

It has been already stated that the structure of GO can be described in terms of a directed acyclic graph (DAG), where each GO term is a node and the relationships between the terms are arcs between the nodes. The relationships used in GO are directed. Terms may have more than one parent, and they may be connected to parent terms or root via different relations. To calculate the distance between a pair of terms $\left(t_i, t_j\right)$, Wu and Palmer method is adopted. One of the method's steps is that it finds the common closest parent from all candidate common parents associated with $\left(t_i, t_j\right)$. In other words, Wu and Palmer method finds the parent with the shortest

distance from (t_i, t_j) and if there are more than one such parent, the method selects the one with the maximum distance from the root. Thus, the desired minimum distance $d(t_i, t_j)$ is calculated from (3.2). So, the reason why we select the minimum N_i and N_j and the maximum N, as it has been mentioned before, is to take the shortest distance when there are multiple paths. Some special cases for the common closest parent's selection are shown in the following example in Figure 18. Also, Figure 19 presents how the Wu and Palmer's method works.



Pair of Terms	Common Closest Parent
B,B	B
B,C	A
A,C	A
A,A	A
D,B	B

Figure 18 : Special cases for the selection of the common closest parent.

The **minimum** *between-term* distance aggregation may then be used as an estimate of the GO-based similarity between two genes products g_k and g_m, which is defined as:

$$d(g_k, g_m) = \begin{cases} \underset{i,j}{avg}\left(1 - \dfrac{2 \cdot N}{N_{ki} + N_{mj} + 2 \cdot N}\right) & \text{if } N \neq 0 \\ \underset{i,j}{avg}\left(1 - \dfrac{2}{N_{ki} + N_{mj}}\right) & \text{if } N = 0 \end{cases} . \quad (3.3)$$

So, for genes products g_k and g_m that are very close each other, it holds that $d(g_k, g_m) \to 0$. However for genes products g_k and g_m that are far away each other, it holds that $d(g_k, g_m) \to 1$. This justifies the assumption in (3.1) when $N = 0$. The above GO-based similarity between two

genes products g_k and g_m, i.e. $d(g_k, g_m)$, represents their biological distance based on the GO knowledge, as calculated via Wu and Palmer's method described above.

Another method to measure similarity between genes products based on the GO would be an information content technique defined by Resnik [22]. This technique consists of determining the amount of information they share in common. This type of methods exploits the assumption that the more information two terms share in common, the more similar they are. An alternative information-theoretic technique was proposed by Lin [26].

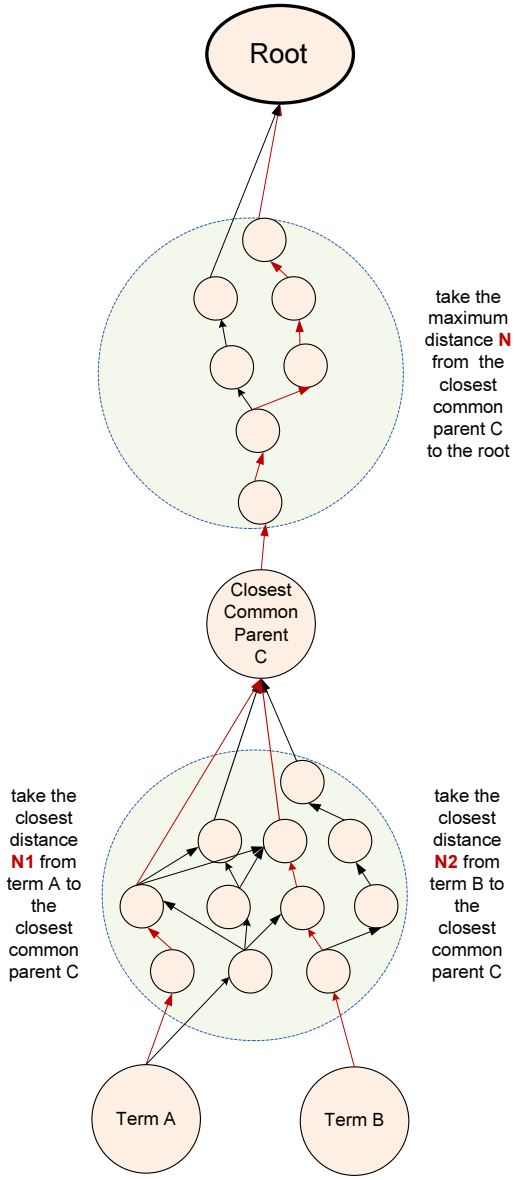


Figure 19 : An example that shows how Wu and Palmer method works.

In this research we implement two hierarchy-specific similarity assessment techniques, each based on information individually extracted from each GO hierarchy (BP or MF), i.e. these techniques are based on the calculation of similarity values, independently obtained from each of the two GO hierarchies. Due to the high computational complexity of Wu and Palmer's method, we keep from the BP or MF hierarchy five terms per gene that appear most frequently. It is important to note that from the full set of genes' terms, almost 37% of genes have more than five annotation terms to BP and almost 40% have more than five annotation terms to MF hierarchy. Concerning the BP hierarchy, approximately the mean number of annotation terms per gene is six with standard deviation five. Almost the same situation exists for MF hierarchy, where the mean is almost five terms per gene with standard deviation almost three. We summarize the above observations in Figure 20. Thus, we infer that by keeping from the BP or MF hierarchy five terms per gene, we lose some of the available biological knowledge and this might affect our results. Additionally, we do not use the CC hierarchy at all, because its relationships are of type " *part of* " and not of type "*is a*", as required in the implementation of Wu and Palmer's method.

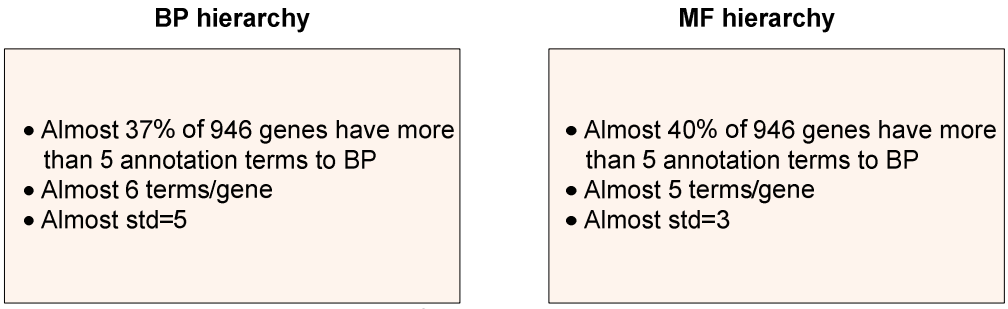


Figure 20 : Some useful statistics about the available genes.

Furthermore, we study an approach based on the aggregation of similarity information originated from both BP and MF hierarchies. These overall GO-based similarity values are calculated by taking into account the combined annotations originated from both GO hierarchies. In this case, a "fake" term is added as a parent of the two existing root nodes from BP and MF hierarchy. Due to the method's remarkable computational complexity, we keep for each gene six annotation terms to the combined BP and MF hierarchy, i.e. we keep three terms that appear most frequently concerning the BP hierarchy and three terms that appear most frequently concerning the MF hierarchy. Thus, according to Figure 20 some of the available biological knowledge is lost, which might affect the results.

It is important to mention that in this thesis, we focus on a method for gene clustering and cluster validity indices, using GO-driven similarity. So, the above GO-based biological distances calculated via Wu and Palmer's method, are used in clustering algorithm and cluster validation. As stated in Chapter 1, incorporating prior knowledge in the clustering process leads to clusters that are more refined and biologically relevant.

3.4 Clustering Method: Hard C-Means

3.4.1 Hard C-Means

K-Means (or C-Means) methodology is a commonly used clustering technique. This is a method of cluster analysis which aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean. In this analysis the user starts with a collection of samples and attempts to group them into k number of clusters based on certain specific distance measurements. The prominent steps involved in the k-means clustering algorithm are given below.

1. This algorithm is initiated by creating k different clusters. The given sample set is first randomly distributed among these k different clusters.
2. The center (centroid) of each cluster is calculated.
3. As a next step, the distance measurement between each of the sample, within a given cluster, to their respective cluster centroid is calculated.
4. Samples are then moved to a cluster that records the shortest distance from a sample to the cluster centroid.
5. If clusters have changed then go to step 2.

The algorithm stops when the clusters become stable (e.g. zero reallocations) or some maximum number of iterations has been performed. This latter test case is necessary because the clusters may not stabilize in a reasonable amount of time for some sets of points. This issue may become more prevalent when dealing with points in higher dimensional spaces and/or when using non-Euclidean distance metrics such as Manhattan distance. Unstable clusters are typically not a

significant issue, since usually even these clusters will be distinct and well formed after just a few iterations.

As a first step to the cluster analysis, the user decides the number of clusters k. This parameter could take definite integer values with the lower bound of 1 (in practice, 2 is the smallest relevant number of clusters) and an upper bound that equals the total number of samples. The k-means algorithm is repeated a number of times to obtain an optimal clustering solution, every time starting with a random set of initial clusters.

In this study, the hard c-means clustering method is applied. In particular, in the classical hard c-means model each data point $\mathbf{x}_i$, which is a vector, in the given data set $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}, X \subseteq R^p$ is assigned to exactly one cluster. Each cluster Γ_i is thus a subset of the given data set, $\Gamma_i \subset X$. The set of clusters $\Gamma = \{\Gamma_1, \ldots \Gamma_c\}$ is required to be an exhaustive partition of the data set X into c non-empty and pairwise disjoint subsets $\Gamma_i, 1 \prec c \prec n$. In the hard c-means such a data partition is said to be optimal when the sum of the squared distances between the cluster centers and the data points assigned to them is minimal [27]. This definition follows directly from the requirement that clusters should be as homogeneous as possible. Hence the objective function of the hard c-means can be written as follows:

$$J_h(X, U_h, C) = \sum_{i=1}^{c} \sum_{j=1}^{n} u_{ij} d_{ij}^2, \tag{3.4}$$

where $C = \{C_1, \ldots, C_c\}$ is the set of cluster prototypes, d_{ij} is the distance between $\mathbf{x}_j$ and cluster center $\mathbf{c}_i$, and U is a $c \times n$ binary matrix called partition matrix. The individual elements

$$u_{ij} \in \{0,1\} \tag{3.5}$$

indicate the assignment of data to clusters: $u_{ij} = 1$ if the data point $\mathbf{x}_j$ is assigned to prototype C_i, i.e. $\mathbf{x}_j \in \Gamma_i$, and $u_{ij} = 0$ otherwise. To ensure that each data point is assigned exactly to one cluster, it is required that:

$$\sum_{i=1}^{c} u_{ij} = 1, \forall j \in \{1, \ldots, n\}. \tag{3.6}$$

This constraint excludes the trivial solution when minimizing J_h, which is that no data is assigned to any cluster: $u_{ij} = 0, \forall i, j$ [12]. Considering (3.6) and the fact that $u_{ij} \in \{0,1\}$ it is impossible for data to be assigned to more than one clusters. However, unfortunately there are some remaining clusters left empty. Since such a situation is undesirable, one usually requires that [12]:

$$\sum_{j=1}^{n} u_{ij} > 0, \forall i \in \{1,\ldots,c\}. \tag{3.7}$$

The objective function J_h depends on two (disjoint) parameter sets, which are the cluster centers c and the assignment of data points to clusters U. The problem of finding parameters that minimize the c-means objective function is NP-hard [28]. Thus, we implement an approach of the hard c-means clustering algorithm that provides the partition by optimizing such a criterion.

In c-means, the parameters to be optimized are split into two (or even more) groups. Then one group of parameters (e.g. the partition matrix) is optimized holding the other group(s) (e.g. the current cluster centers) fixed (and vice versa). This iterative updating scheme is then repeated. The main advantage of this method is that in each of the steps the optimum can be computed directly. By iterating the two (or more) steps the joint optimum is approached, although it cannot be guaranteed that the global optimum will be reached. The algorithm may get stuck in a local minimum of the applied objective function J. However, alternating optimization is the commonly used parameter optimization method in clustering algorithms.

In case of hard c-means, the iterative optimization scheme works as follows: at first, initial cluster centers are chosen. This can be done randomly, i.e. by picking c random vectors that lie within the smallest (hyper-)box that encloses all data or by initializing cluster centers with randomly chosen data points of the given data set. In this study, initial cluster centers are chosen via the second approach. Alternatively, more sophisticated initialization methods can be used as well, e.g. Latin hypercube sampling [31]. Then the parameters C are held fixed and cluster assignments U are determined that minimize the quantity of J_h. In this step each data point is assigned to its closest cluster center:

$$u_{ij} = \begin{cases} 1, & \text{if } i = \arg\min_{l=1,\ldots,c} d_{lj} \\ 0, & \text{otherwise} \end{cases}. \tag{3.8}$$

Any other assignment of a data point would not minimize J_h for fixed clusters. Then the data partition U is held fixed and new cluster centers are computed as the mean of all data vectors assigned to them, since the mean minimizes the sum of the square distances in J_h. The calculation of the mean for each cluster (for which the algorithm got its name) is stated more formally:

$$\mathbf{c}_i = \frac{\sum_{j=1}^{n} u_{ij}\mathbf{x}_j}{\sum_{j=1}^{n} u_{ij}}. \tag{3.9}$$

The two steps (3.8) and (3.9) are iterated until no change in C or U can be observed. Then the hard c-means terminates, yielding final cluster centers and gene partition that are possibly only locally optimal.

The hard c-means algorithm is fast, since at each iteration $c \cdot n$ dissimilarities are evaluated and c centroids are updated. This fact makes hard c-means a popular algorithm, allowing it to cluster thousands of objects. That is why we applied this clustering method in our research.

Concluding the presentation of the hard c-means, it is important to mention its expressed tendency to become stuck in local minima, which makes it necessary to conduct several runs of the algorithm with randomly different initializations [30]. Then the best result out of many clusterings can be chosen based on the values of J_h. So, in this research we conduct 10 runs of the hard c-means with randomly different initializations and we choose as the best result the one with the min value of J_h. Stochastic optimal search techniques, such as simulated annealing and genetic algorithms, provide a possible way to search the complicated problem space more effectively and find the global or approximately global optimum.

Another disadvantage of this algorithm is that hard c-means assumes that the number of clusters c is already known by the users, which unfortunately often is not true in practice. Like the situation for cluster initialization mentioned above, there are also no efficient and universal methods for the selection of c. Therefore, identifying c in advance becomes a very important topic

in cluster validity. There are several heuristics that are directly related to hard c-means. An example is the ISODATA (Iterative Self - Organizing Data Analysis Technique) algorithm [32] which deals with the dynamic estimation of c. Moreover, in order to estimate the number of clusters, c, in a data set, a variety of validity measures exists. More details about validity measures are discussed in Section 3.6.

Furthermore, c-means is sensitive to outliers and noise. The calculation of the means considers all the data objects in the cluster, including the outliers. Even if an object is quite far away from the cluster centroids, it is still forced into a cluster and used to calculate the prototype representation, which therefore distorts the cluster shapes. So, there are some methods which handle this lack of robustness. For example ISODATA [32] and PAM (Partitioning Around Medoids) [33] both consider the effect of outliers in clustering procedures. ISODATA discards the clusters in which the number of data points is below some threshold. It splits a cluster if the within-cluster variability is above a threshold, or combines two clusters if their prototypes are close enough (judged by another threshold). The disadvantage of this approach is the need for the user to select the parameters. The splitting operation of ISODATA eliminates the possibility of elongated clusters typical of c-means. Also, as far as PAM is concerned, rather than utilizing the calculated means, PAM utilizes real data points, called medoids, as the cluster prototypes, and it avoids the effect of outliers to the resulting prototypes. A medoid is a point that has the minimal average distance to all other objects in the same cluster.

Finally, the definition of means limits the application of c-means only to numerical variables, while leaving the categorical variables unhandled. Moreover, even for the numerical variables, the obtained means may not have the physical meaning or may be difficult to interpret. The author in [34] discussed hard c-means in binary data clustering and suggested three variants. It is indicated that binary data can also be used to represent categorical data. The authors in [35] and [36] defined different dissimilarity measures to extend c-means to categorical variables.

More recent discussions on hard c-means, its variants, and other squared-error based clustering algorithms with their applications can be found, for example in [37] and [38].

3.4.2 Distance Metrics

Distance Types

The following distance types can be used for clustering [71].

- *Euclidean distance (L2-norm)*: This is the most usual, "natural" and intuitive way of computing a distance between two samples. It takes into account the difference between two samples directly, based on the magnitude of changes in the sample levels. This distance type is usually used for data sets that are suitably normalized or without any special distribution problem.

- *Manhattan distance (L1-norm)*: Also known as city-block distance, this distance measurement is especially relevant for discrete data sets. While the Euclidean distance corresponds to the length of the shortest path between two samples, the Manhattan distance refers to the sum of distances along each dimension.

- *Pearson Correlation distance*: This distance is based on the Pearson correlation coefficient that is calculated from the sample values and their standard deviations. The correlation coefficient r takes values from -1 (large, negative correlation) to $+1$ (large, positive correlation). Effectively, the Pearson distance dp is computed as $dp = 1 - r$ and lies between 0 (when correlation coefficient is $+1$, i.e. the two samples are most similar) and 2 (when correlation coefficient is -1). Note that the data are centered by subtracting the mean and scaled by dividing by the standard deviation.

- *Absolute Pearson Correlation distance*: In this distance, the absolute value of the Pearson correlation coefficient is used, hence the corresponding distance lies between 0 and 1, just like the correlation coefficient. The Absolute Pearson distance da is given as $da = 1 - \frac{1}{2} \cdot r^{\frac{1}{2}}$. Taking the absolute value gives equal meaning to positive and negative correlations, due to which anti-correlated samples will get clustered together.

- *Un-centered Correlation distance*: This is the same as the Pearson correlation, except that the sample means are set to zero in the expression for un-centered correlation. The un-

centered correlation coefficient lies between -1 and $+1$, hence the distance lies between 0 and 2.

- *Absolute Un-centered Correlation distance*: This is the same as the Absolute Pearson correlation, except that the sample means are set to zero in the expression for un-centered correlation. The un-centered correlation coefficient lies between 0 and $+1$, hence the distance lies between 0 and 1.

- *Kendall's (tau) distance*: This non-parametric distance measurement is more useful in identifying samples with a huge deviation in a given data set.

The standard distance metric used with k-means is Euclidean (L2-norm) distance. However, the justification for using L2 distance is not always clear. When points are composed of multiple independent (or mostly independent) variables, there is a case for expecting Manhattan (L1-norm) distance to be a better measure of distance between two points. In this study, Euclidean distance (the default distance measure for c-means) has been used in hard c-means method based only on gene expression values. The Euclidean distance can sometimes be misleading. So, domain knowledge must be used to guide the formulation of a suitable distance measure for each particular application. Thus, despite the fact that Euclidean distance is sensitive to high values, this distance is widely used in the analysis of gene expression data.

3.4.3 Variations of Hard C-Means

Furthermore, as far as clustering methods implemented in this thesis are concerned, in order to find the gene clusters based only on KEGG knowledge, three approaches of the hard c-means clustering method are implemented. At first, we create the partition vector for each gene based on the available KEGG knowledge, i.e. based on a vector whose elements take value 0 if the gene does not take part in a specific pathway or 1 if the gene does take part in this specific pathway. These vectors are the columns of a partition matrix $196x946$, where 196 is the number of the discrete pathways and 946 is the number of the discrete genes. Next, we apply the hard c-means to this partition matrix using the Euclidean and the Correlation distance metrics. It is noted that in order the Correlation distance metric to take values from -1 (large, negative correlation) to $+1$ (large, positive correlation), we normalize the data to unit norm. It is also important

to note that the approach using the Euclidean distance metric aims to minimize the objective function J_h, while the approach using the Correlation distance metric aims to maximize the objective function J_h. For better results, we conduct 10 runs of the above approaches with randomly different initializations and we choose as the best result the one with the min or max value of J_h respectively. It is noted that the two above distance metrics, i.e. the Euclidean and the Correlation distance metrics are the same with the distance metrics that the classical hard c-means clustering method use. The only difference is that the above distance metrics are applied on partition vectors obtained from the available KEGG knowledge, as it has been presented before.

In addition, we implement another approach of hard c-means clustering method in order to find the biological clusters based only on KEGG knowledge, i.e. based only on the aforementioned partition matrix. The basic idea in this approach of hard c-means clustering algorithm is similar to the previous implementations, but with some differences. Firstly, initial cluster centers are chosen. This is done randomly, i.e. by initializing cluster centers with randomly chosen genes of the given data set. Then the parameters C are held fixed and cluster assignments U are determined that maximize the quantity of $J_h^{'}$, where now the objective function is the $J_h^{'}\left(X,U_h,C\right)=\sum_{i=1}^{c}\sum_{j=1}^{n}u_{ij}d_{ij}^{'}$, where $d_{ij}^{'}$ indicates the number of common pathways between the gene j and the cluster center i. In this step each gene is assigned to a cluster so as the gene to have the maximum number of common pathways with that cluster center. Any other assignment of a gene would not maximize $J_h^{'}$ for fixed clusters. Then the gene partition U is held fixed and new cluster centers are computed as follows: the partition vector of the cluster center takes the value 0 if the number of genes in this cluster that have 0 in the corresponding position of their partition vectors is greater than the number of genes that have 1 or vice versa. An illustrative example of a cluster with three genes is shown in Table 1. The above two steps are again iterated until no change in C or U can be observed. Also, in each iteration except for the last, when a cluster becomes empty then we force a random gene to belong to that cluster. When this approach of hard c-means terminates, it yields final cluster centers and gene partition that are possibly only locally optimal. For better results, we again conduct 10 runs of this approach with randomly different initializations and we choose as the best result the one with the max value of

$J_h^{'}$. The performance of this algorithm is satisfactory. This will be also justified in the next chapter which presents the performance of all implemented algorithms in detail. Furthermore, as far as the third approach is concerned, we can see that this approach is similar to c-means, but now the approach does not use the mean but the $d_{ij}^{'}$ distance metric in order to distribute the genes among the clusters.

	Gene1	**Gene2**	**Gene3**	**Cluster Center**
Partition Vectors	0	1	0	0
	0	0	0	0
	1	1	0	1
	1	1	1	1

Table 1 : An illustrative example that shows the calculation of a cluster center's partition vector.

Finally, in order to find the gene clusters based only on GO knowledge another approach of hard c-means clustering method has been implemented in this thesis too. The basic idea in this approach of hard c-means clustering algorithm is similar to the previous implementations, but with some differences. Firstly, initial cluster centers are chosen. This is done randomly, i.e. by initializing cluster centers with randomly chosen genes of the given data set. Then the parameters C are held fixed and cluster assignments U are determined that minimize the quantity of $J_h^{''}$, where now the objective function is the $J_h^{''}(X, U_h, C) = \sum_{i=1}^{c}\sum_{j=1}^{n} u_{ij} d_{ij}^{''}$, where $d_{ij}^{''}$ indicates the biological distance[14] between the gene j and the cluster center i. In this step each gene is assigned to its closest cluster so as the gene to have the minimum biological distance with the cluster center. Any other assignment of a gene would not minimize $J_h^{''}$ for fixed clusters. Then the gene partition U is held fixed and new cluster centers are computed as follows: for each cluster, its

[14] Biological distances, which are based on the GO hierarchies, are calculated via the Wu and Palmer's method, described in Section 3.3.

center becomes the gene that has the minimum sum of squared biological distances with all the genes belong to that cluster. It is important to note that this criterion of choosing the appropriate cluster center is the same criterion with that used in the classical hard c-means model, i.e. the calculation of the mean for each cluster. It is like taking the nearest neighbor to mean, which obtained in classical hard c-means from equation (3.9). The only difference now is that the solution is one of the available vectors (genes) for each cluster. So, from all above the choice of this GO-based approach's criterion is not arbitrary. The above two steps are again iterated until no change in C or U can be observed. Then this approach of hard c-means terminates, yielding final cluster centers and gene partition that are possibly only locally optimal. For better results, we again conduct 10 runs of this approach with randomly different initializations and we choose as the best result the one with the min value of $J_h^{''}$. The performance of this algorithm is satisfactory. This will be also justified in the next chapter which presents the performance of all implemented algorithms in detail.

Figure 21 presents the different approaches of hard c-means clustering method implemented in this thesis, as they have been mentioned above.

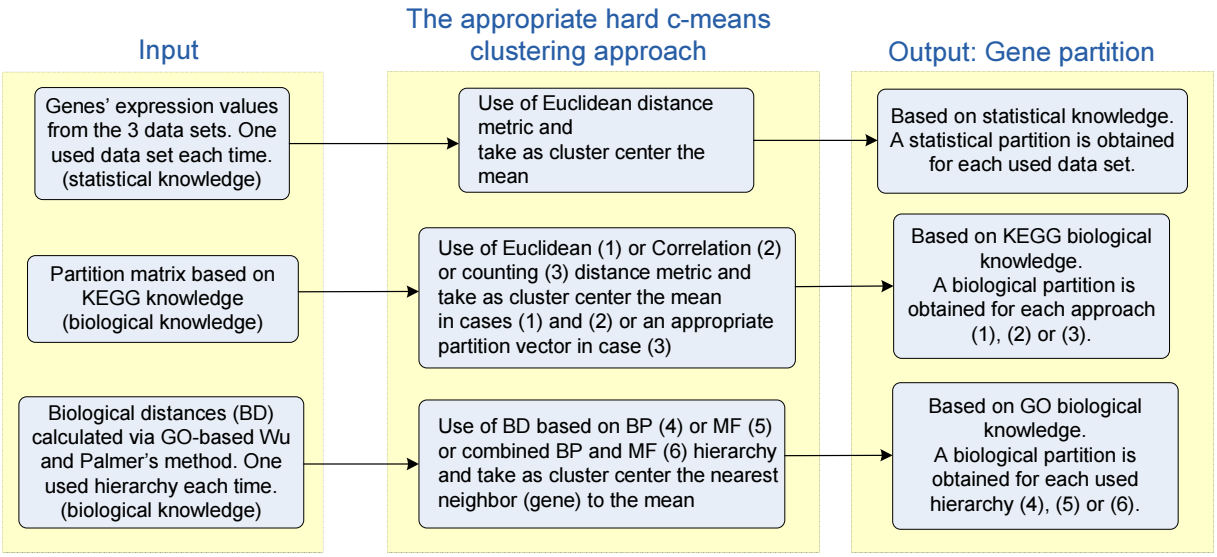


Figure 21 : The different approaches of hard c-means clustering method implemented.

3.5 Fuzzy C-Means (FCM)

Another approach of cluster analysis is fuzzy cluster analysis. This approach allows gradual memberships of data points to clusters measured as degrees in [0,1]. Thus, it gives the flexibility to express that data points can belong to more than one cluster. Furthermore, these membership degrees offer a much finer degree of detail of the data model. Aside from assigning a data point to clusters in shares, membership degrees can also express how ambiguously or definitely a data point should belong to a cluster. The concept of these membership degrees is substantiated by the definition and interpretation of fuzzy sets [63]. Thus, fuzzy clustering allows fine grained solution spaces in the form of fuzzy partitions of the set of given examples $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$. Whereas the clusters Γ_i of data partitions have been classical subsets so far, they are represented by the fuzzy sets μ_{Γ_i} of the data-set X in the following. Complying with fuzzy set theory, the cluster assignment u_{ij} is now the membership degree of a datum $\mathbf{x}_j$ to cluster Γ_i, such that: $u_{ij} = \mu_{\Gamma_i}(\mathbf{x}_j) \in [0,1]$. Since memberships to clusters are fuzzy, there is not a single label that is indicating to which cluster a data point belongs. Instead, fuzzy clustering methods associate a fuzzy label vector to each data point $\mathbf{x}_j$ that states its memberships to the c clusters:

$$\mathbf{u}_j = (u_{1j}, \ldots, u_{cj})^T. \tag{3.10}$$

The $c \times n$ matrix $U = (u_{ij}) = (\mathbf{u}_1, \ldots, \mathbf{u}_n)$ is then called a fuzzy partition matrix. Based on the fuzzy set notion we are now better suited to handle ambiguity of cluster assignments when clusters are badly delineated or overlapping.

So far, the general definition of fuzzy partition matrices leaves open how assignments of data to more than one cluster should be expressed in form of membership values. Furthermore, it is still unclear what degrees of belonging to clusters are allowed, i.e. the solution space (set of allowed fuzzy partitions) for fuzzy clustering algorithms is not yet specified. In the field of fuzzy clustering two types of fuzzy cluster partitions have evolved. They differ in the constraints they place on the membership degrees and how the membership values should be interpreted. In this Section we discuss about the most widely used type, the probabilistic partitions, since they have been proposed first. Notice, that in literature they are sometimes just called fuzzy partitions

(dropping the word "probabilistic"). We use the subscript f for the probabilistic approaches. The second type of fuzzy partitions is the possibilistic models. The subscript p is used for these methods. More details about the possibilistic models will be discussed in the end of this Section.

As far as the first type of fuzzy partitions is concerned, let $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ be the set of given examples and let c be the number of clusters $(1 < c < n)$ represented by the fuzzy sets μ_{Γ_i}, $(i = 1, \ldots, c)$. Then we call $U_f = (u_{ij}) = (\mu_{\Gamma_i}(\mathbf{x}_j))$ a *probabilistic cluster partition* of X if

$$\sum_{j=1}^{n} u_{ij} > 0, \ \forall i \in \{1, \ldots, c\}, \quad \text{and} \tag{3.11}$$

$$\sum_{i=1}^{c} u_{ij} = 1, \ \forall i \in \{1, \ldots, n\} \tag{3.12}$$

hold. The $u_{ij} \in [0,1]$ are interpreted as the membership degree of datum $\mathbf{x}_j$ to cluster Γ_i relative to all other clusters.

Constraint (3.11) guarantees that no cluster is empty. This corresponds to the requirement in classical cluster analysis that no cluster, represented as (classical) subset of X, is empty (see Equation (3.7) in Subsection 3.4.1). Condition (3.12) ensures that the sum of the membership degrees for each datum equals 1. This means that each datum receives the same weight in comparison to all other data and, therefore, that all data are (equally) included into the cluster partition. This is related to the requirement in classical clustering that partitions are formed exhaustively (see Equation (3.6) in Subsection 3.4.1). As a consequence of both constraints no cluster can contain the full membership of all data points. Furthermore, condition (3.12) corresponds to a normalization of the memberships per datum. Thus the membership degrees for a given datum formally resemble the probabilities of its being a member of the corresponding cluster.

After defining probabilistic partitions we can turn to developing an objective function for the fuzzy clustering task. Certainly, the closer a data point lies to the center of a cluster, the higher its degree of membership should be to this cluster. Following this rationale, one can say that the distances between the cluster centers and the data points should be minimal. Hence the problem to divide a given data set into c clusters can be stated as the task to minimize the squared distances of the data points to their cluster centers, since, of course, we want to maximize the de-

grees of membership. The probabilistic fuzzy objective function J_f is thus based on the least sum of squared distances, just as J_h of the hard c-means, presented in Subsection 3.4.1. More formally, a fuzzy cluster model of a given data-set X into c clusters is defined to be optimal when it minimizes the objective function:

$$J_f\left(X, U_f, C\right) = \sum_{i=1}^{c}\sum_{j=1}^{n} u_{ij}^m d_{ij}^2 , \tag{3.13}$$

under the constraints (3.11) and (3.12) that have to be satisfied for probabilistic membership degrees in U_f. The condition (3.11) avoids the trivial solution of minimization problem, i.e. $u_{ij} = 0,\ \forall i, j$. The normalization constraint (3.12) leads to a "distribution" of the weight of each data point over the different clusters. Since all data points have the same fixed amount of membership to share between clusters, the normalization condition implements the known partitioning property of any probabilistic fuzzy clustering algorithm. The parameter m, $m > 1$, is called the *fuzzifier* or *weighting exponent*. The exponentiation of the memberships with m in J_f can be seen as a function g of the membership degrees, $g\left(u_{ij}\right) = u_{ij}^m$, that leads to a generalization of the well-known least squared error functional as it was applied in the hard c-means (see Equation (3.4) in Subsection 3.4.1). The actual value of m then determines the "fuzziness" of the classification. It has been shown for the case $m = 1$ (when J_h and J_f become identical), that cluster assignments remain hard when minimizing the target function, even though they are allowed to be fuzzy, i.e. even though they are not constrained in $\{0,1\}$ [64]. For achieving the desired fuzzification of the resulting probabilistic data partition the function $g\left(u_{ij}\right) = u_{ij}^2$ has been proposed first [64]. The generalization for exponents $m > 1$ that lead to fuzzy memberships has been proposed in [65]. With higher values for m the boundaries between clusters become softer, with lower values they get harder. Usually $m = 2$ is chosen. Aside from the standard weighting of the memberships with u_{ij}^m other functions g that can serve as fuzzifiers, have been explored.

The objective function J_f is alternately optimized, i.e. first the membership degrees are optimized for fixed cluster parameters, then the cluster prototypes are optimized for fixed membership degrees:

$$U_\tau = j_U\left(C_{\tau-1}\right),\ \tau > 0 \quad \text{and} \tag{3.14}$$

$$C_\tau = j_C\left(U_\tau\right). \tag{3.15}$$

In each of the two steps the optimum can be computed directly using the parameter update equations j_U and j_C for the membership degrees and the cluster centers, respectively. The update formulae are derived by simply setting the derivative of the objective function J_f with regard to the parameters to optimize equal to zero (taking into account the constraint (3.12)). The resulting equations for the two iterative steps form the fuzzy c-means algorithm.

The membership degrees have to be chosen according to the following update formula that is independent of the chosen distance measure [66], [67]:

$$u_{ij} = \frac{1}{\sum_{l=1}^{c}\left(\frac{d_{ij}^2}{d_{lj}^2}\right)^{\frac{1}{m-1}}} = \frac{d_{ij}^{\frac{2}{m-1}}}{\sum_{l=1}^{c} d_{lj}^{\frac{2}{m-1}}}. \tag{3.16}$$

In this case there exists a cluster i with zero distance to a datum $\mathbf{x}_j$, $u_{ij} = 1$ and $u_{lj} = 0$ for all other clusters $l \neq i$. The above equation clearly shows the relative character of the probabilistic membership degree. It depends not only on the distance of the datum $\mathbf{x}_j$ to cluster i, but also on the distances between this data point and other clusters.

The update formulae j_C for the cluster parameters depend, of course, on the parameters used to describe a cluster (location, shape, size) and on the chosen distance measure. Therefore a general update formula cannot be given. In the case of the basic fuzzy c-means model the cluster center vectors serve as prototypes, while an inner product norm induced metric is applied as distance measure. Consequently the derivations of J_f with regard to the centers yield [66]:

$$\mathbf{c}_i = \frac{\sum_{j=1}^{n} u_{ij}^m \mathbf{x}_j}{\sum_{j=1}^{n} u_{ij}^m}. \tag{3.17}$$

The choice of the optimal cluster center points for fixed memberships of the data to the clusters has the form of a generalized mean value computation for which the fuzzy c-means algorithm has its name.

The general form of the optimized scheme of coupled equations (3.14) and (3.15) starts with an update of the membership matrix in the first iteration of the algorithm ($\tau = 1$). The first calculation of memberships is based on an initial set of prototypes C_0. Even though the optimization of an objective function could mathematically also start with an initial but valid membership matrix (i.e. fulfilling constraints (3.11) and (3.12)), a C_0 initialization is easier and therefore common practice in all fuzzy clustering methods. Basically the fuzzy c-means can be initialized with cluster centers that have been randomly placed in the input space. The repetitive updating in the optimized scheme can be stopped, if the number of iterations τ exceeds some predefined number of maximal iterations $\tau_{\max}$, or when the changes in the prototypes are smaller than some termination accuracy. The (probabilistic) fuzzy c-means algorithm is known as a stable and robust classification method. Compared with the hard c-means, presented in Subsection 3.4.1, it is quite insensitive to its initialization and it is not likely to get stuck in an undesired local minimum of its objective function in practice [68]. Due to its simplicity and low computational demands, the probabilistic fuzzy c-means is a widely used initializer for other more sophisticated clustering methods. On the theoretical side it has been proven that either the iteration sequence itself or any convergent subsequence of the probabilistic FCM converges in a saddle point or a minimum – but not in a maximum – of the objective function [66].

Although often desirable, the "relative" character of the probabilistic membership degrees can be misleading [69]. Fairly high values for the membership of datum in more than one cluster can lead to the impression that the data point is typical for the clusters, but this is not always the case. For a correct interpretation of these memberships one has to keep in mind that they are rather degrees of sharing than of typicality, since the constant weight of 1 given to a datum must be distributed over the clusters. A better reading of the memberships, avoiding misinterpretations, would be [70]: "If the datum $\mathbf{x}_i$ has to be assigned to a cluster, then with the probability u_{ij} to the cluster i".

The normalization of memberships can further lead to undesired effects in the presence of noise and outliers. The fixed data point weight may result in high membership of these points to clusters, even though they are a large distance from the bulk of data. Their membership values consequently affect the clustering results, since data point weight attracts cluster prototypes. By dropping the normalization constraint (3.12), the possibilistic models try to achieve a more intuitive assignment of degrees of membership and to avoid undesirable normalization effects. More information about the possibilistic models can be found in [12].

3.6 Cluster Validity Indices

The prediction of the correct number of clusters is a fundamental problem in unsupervised classification problems. Many clustering algorithms require the definition of the number of clusters beforehand. To overcome this problem, various cluster validity indices have been proposed to assess the quality of a clustering partition. This approach requires the execution of a clustering algorithm several times to obtain different partitions. The clustering partition that optimizes a validity index is selected as the best partition. Thus, the main goal of a cluster validity technique is to identify the partition of clusters for which a measure of quality is optimal.

Cluster validity measures are used to compare different partitions created by different clustering algorithms, or by the same algorithm using different parameter values. Cluster validation is very important issue in clustering analysis because the result of clustering needs to be validated in most applications. In most clustering algorithms, the number of clusters is set as user parameter. There are a lot of approaches to find the best number of clusters. A variety of validity measures are available and so it is possible to find out: first, how well clustering algorithms have worked and how altering parameters effects the clustering and second, the similarity between the validity measures.

In this study, cluster validation is performed using four validity measures: the C-index, the Goodman-Kruskal index, the Dunn index and the Silhouette index. These validity methods have been shown to be efficient cluster validity estimators for different types of clustering applications. Furthermore, they have been chosen to support the investigation of cluster validation tech-

niques for genome expression data classification. Nevertheless, each of the implemented validation methods has its advantages and limitations.

Basic Distance Metrics

The distance between two samples $\mathbf{x}$ and $\mathbf{y}$, which are vectors, in the data set for interval type of the data[15], $d(\mathbf{x},\mathbf{y})$, in all validity measures was calculated using the well-known *Euclidean*, *Manhattan* and *Chebychev* metrics [2]:

- *Euclidean Distance:* $d(\mathbf{x},\mathbf{y}) = \sqrt{\sum(\mathbf{x}-\mathbf{y})^2}$
- *Manhattan Distance:* $d(\mathbf{x},\mathbf{y}) = \sum|\mathbf{x}-\mathbf{y}|$
- *Chebychev Distance:* $d(\mathbf{x},\mathbf{y}) = \max|\mathbf{x}-\mathbf{y}|$

The **C-index** [14], C, is defined as follows:

$$C = \frac{S - S_{\min}}{S_{\max} - S_{\min}}, \tag{3.18}$$

where S, $S_{\min}$, $S_{\max}$ are calculated as follows. Let p be the number of all pairs of samples (conditions) from the same cluster. Then S is the sum of distances between samples in those p pairs. Let P be a number of all possible pairs of samples in the dataset. Ordering those P pairs by distances we can select p pairs with the smallest and p pairs with the largest distances between samples. The sum of the p smallest distances is equal to $S_{\min}$, whilst the sum of the p largest is equal to $S_{\max}$. From this formula it follows that the nominator will be small if pairs of samples with small distances are in the same cluster. Thus, small values of C correspond to good clusters. We calculate distances using the data-driven (using Euclidean, Manhattan and Chebychev metrics) and knowledge-driven (using biological distances calculated via GO-based Wu and Palmer's method) methods. The number of clusters that minimize C-index is taken as the optimal number of clusters, c. We implement two approaches of C-index. From all above, it is noted that for each cluster in a partition, a C-index is calculated according to the aforemen-

[15] Interval data (also sometimes called integer) is measured along a scale in which each position is equidistant from one another. This allows for the distance between two pairs to be equivalent in some way. Interval data cannot be multiplied or divided. For example: temperature in degrees Fahrenheit.

tioned strategy. Thus, the first approach selects the minimal C-index from a set of candidate C-indices. These candidates obtain by keeping in each number of clusters (i.e. in each partition) the maximum C-index (the worst case) from all its calculated clusters' C-indices. As far as the second approach is concerned, this approach selects the minimal C-index from a set of candidate C-indices. These candidates obtain by keeping in each number of clusters (i.e. in each partition) the sum of all its calculated clusters' C-indices. The C-index is an effective cluster validity estimator for different types of clustering applications.

The **Goodman-Kruskal index** [15], GK, is another validity measure. For a given dataset, $\mathbf{X}_j, j = 1,\ldots,k$, where k is the total number of samples (gene products in this application) in the dataset, this method assigns all possible quadruples [20]. Let d be the distance between any two samples ($\mathbf{w}$ and $\mathbf{x}$, or $\mathbf{y}$ and $\mathbf{z}$, where $\mathbf{w}$, $\mathbf{x}$, $\mathbf{y}$ and $\mathbf{z}$ are all vectors) in $\mathbf{X}_j$. A quadruple is called concordant if one of the following two conditions is true:

- $d(\mathbf{w},\mathbf{x}) < d(\mathbf{y},\mathbf{z})$, $\mathbf{w}$ and $\mathbf{x}$ are in the same cluster and $\mathbf{y}$ and $\mathbf{z}$ are in different clusters.
- $d(\mathbf{w},\mathbf{x}) > d(\mathbf{y},\mathbf{z})$, $\mathbf{w}$ and $\mathbf{x}$ are in different clusters and $\mathbf{y}$ and $\mathbf{z}$ are in the same cluster.

By contrast, a quadruple is called disconcordant if one of following two conditions is true:

- $d(\mathbf{w},\mathbf{x}) < d(\mathbf{y},\mathbf{z})$, $\mathbf{w}$ and $\mathbf{x}$ are in different clusters and $\mathbf{y}$ and $\mathbf{z}$ are in the same cluster.
- $d(\mathbf{w},\mathbf{x}) > d(\mathbf{y},\mathbf{z})$, $\mathbf{w}$ and $\mathbf{x}$ are in the same cluster and $\mathbf{y}$ and $\mathbf{z}$ are in different clusters.

A good partition is one with many concordant and few disconcordant quadruples. Let N_{con} and N_{dis} denote the number of concordant and disconcordant quadruples, respectively. Then the Goodman-Kruskal index, GK, is defined as:

$$GK = \frac{N_{con} - N_{dis}}{N_{con} + N_{dis}}. \tag{3.19}$$

Large values of GK are associated with good partitions. We calculate distances using the data-driven (using Euclidean, Manhattan and Chebychev metrics) and knowledge-driven (using biological distances calculated via GO-based Wu and Palmer's method) methods. Thus, the number of clusters that maximize the Goodman-Kruskal index is taken as the optimal number of clusters, c. Goodman-Kruskal index is expected to be robust against outliers because quadruples of pat-

terns are used for its computation. However, its drawback is its high computational complexity in comparison, for example, with the C-index.

The **Silhouette index** [17] is another method for validation of cluster analysis. For a given cluster, $X_j\,(j=1,\ldots,c)$, the silhouette technique assigns to the ith sample of X_j a quality measure, $s(i)$ $(i=1,\ldots,m)$, known as the *silhouette width*. This value is a confidence indicator on the membership of the ith sample in cluster X_j and it is defined as:

$$s(i)=\frac{\left(d_{\min}(i)-\overline{d}(i)\right)}{\max\left\{\overline{d}(i),d_{\min}(i)\right\}}, \tag{3.20}$$

where $\overline{d}(i)$ is the average distance between the ith sample and all of the samples included in X_j and $d_{\min}(i)$ is the minimum average distance between the ith sample and all of the samples clustered in $X_k\,(k=1,\ldots,c;\ \mathrm{k}\neq\mathrm{j})$. From this formula it follows that $-1\le s(i)\le 1$. For a concrete illustration, see Figure 22. When a cluster contains only a single object it is unclear how $\overline{d}(i)$ should be defined, and then we simply set $s(i)$ equal to zero. This choice is of course arbitrary, but a value of zero appears to be most neutral [17]. When $s(i)$ is close to 1, one may infer

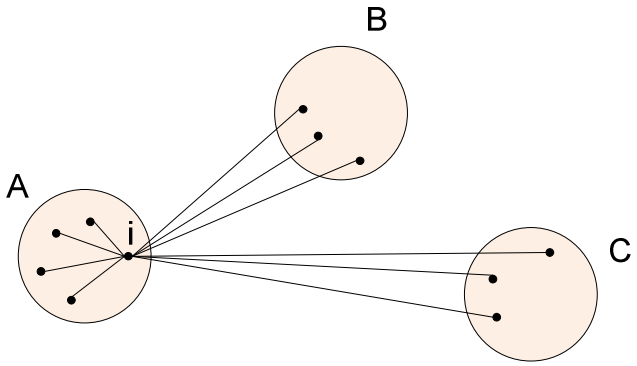


Figure 22 : An illustration of the elements involved in the computation of $s(i)$, where the object i belongs to cluster A.

that the ith sample has been "well-clustered", i.e. it has been assigned to an appropriate cluster. When $s(i)$ is close to zero, it suggests that the ith sample could also be assigned to the nearest

neighbouring[16] cluster, i.e. the ith sample lies equally far away from both clusters, so it can be considered as an "intermediate case". If $s(i)$ is close to −1 (the worst case), one may argue that such a sample has been "misclassified" [22]. To conclude, $s(i)$ measures how well object i matches the clustering at hand (that is, how well it has been classified). So, for each cluster, we can define the average Silhouette width as the average of the $s(i)$ for all objects belonging to that cluster. This allows us to distinguish "clear-cut" clusters with large values of S_j from "weak" clusters with min values of S_j. Thus, for a given cluster X_j, it is possible to calculate a *cluster silhouette* S_j, which characterizes the heterogeneity and isolation properties of such a cluster:

$$S_j = \frac{1}{c}\sum_{i=1}^{m} s(i), \tag{3.21}$$

where m is number of samples in S_j. Moreover, for any partition $U \leftrightarrow X: \mathrm{X}_1 \cup \ldots X_i \cup \ldots X_c$, a *global silhouette value* or *silhouette index,* GSu, can be used as an effective validity index for a partition U:

$$GSu = \frac{1}{c}\sum_{j=1}^{c} S_j . \tag{3.22}$$

Furthermore, it has been demonstrated that equation (3.22) can be applied to estimate the most appropriate number of clusters for partition U. We calculate distances using the data-driven (using Euclidean, Manhattan and Chebychev metrics) and knowledge-driven (using biological distances calculated via GO-based Wu and Palmer's method) methods. In this case the partition with the maximum silhouette index, GSu, is taken as the optimal partition.

Silhouette index offers the advantage that it only depends on the actual partition of the objects and not on the clustering algorithm that was used to obtain it [17]. As a consequence, silhouettes could be used to improve the results of cluster analysis (for instance by moving an ob-

[16] This is like the second best choice for object i: if it could not be accommodated into cluster A, it will be assigned to cluster B, which is the closest competitor to A.

ject with negative $s(i)$ to its neighbor), or to compare the output of different clustering algorithms applied to the same data.

The **Dunn index** [18] is the last validity measure we apply. The Dunn index defines the ratio between the minimal intercluster distance to maximal intracluster distance. The index is given by:

$$D = \frac{\delta_{\min}}{\Delta_{\max}}, \tag{3.23}$$

where $\delta_{\min}$ denotes the smallest distance between two objects from different clusters and $\Delta_{\max}$ denotes the largest distance of two objects from the same cluster. We calculate distances using the data-driven (using Euclidean, Manhattan and Chebychev metrics) and knowledge-driven (using biological distances calculated via GO-based Wu and Palmer's method) methods. The Dunn index is limited to the interval $[0, \infty]$ and should be maximized.

The Dunn's validity index requires the definition of at least two clusters. The same situation applies to the Silhouette method, since to compute the minimum average distance between the sample in one cluster and all of the samples from different clusters, the Silhouette width formula (3.20) requires at least two clusters. Thus, calculations for null-case are not considered here.

Furthermore, the Dunn index has the disadvantage of over-sensitivity to noise, for which a family of 18 cluster validation indices is proposed based on the different definitions of intercluster and intracluster distance.

Intercluster Distances

Now, we will present the internal measures used in the implementation of the Dunn's validity index. As far as intercluster distances are concerned, there are six intercluster distances are used for the calculation of the Dunn's validity index [19]:

- *Single Linkage*: It is the closest distance between two samples belonging to two different clusters.
- *Complete Linkage*: It represents the distance between the most remote samples belonging to two different clusters.

- *Average Linkage*: It defines the average distance between all of the samples belonging to two different clusters.

- *Centroid Linkage*: It is used only for Euclidean distance. It is the Euclidean distance between the centres of two clusters, as calculated by arithmetic mean.

- *Average of Centroids Linkage*: It reflects the distance between the centre of a cluster and all of samples belonging to a different cluster.

- *Hansdorff Metrics*: They are based on the discovery of a maximal distance from samples of one cluster to the nearest sample of another cluster.

In this study, we implement for intercluster distances the single linkage and the complete linkage.

Intracluster Distances

Also, as far as intracluster distances are concerned, there are three intracluster distances are used to calculate the Dunn's validity index [19]. These are:

- *Complete Diameter*: It defines the distance between the most remote samples belonging to the same cluster.

- *Average Diameter*: It represents the average distance between all of the samples belonging to the same cluster.

- *Centroid Diameter*: It reflects the double average distance between all of the samples and the cluster's centre.

In this study, we implement for intracluster distances the complete diameter and the average diameter.

Based on an external cluster validation the validity measures were evaluated and compared on the basis of various sets of t-invariants of different types of Petri nets (i.e. metabolic, gene regulatory and signal transduction nets). With respect to the percentage of correct predictions best results were obtained using the Silhouette Width (75%) and the C-index (75%), followed by the Dunn-index (50%). Although offering good results, the C-index is hampered by the fact of

showing optimal index values for different numbers of clusters, thus impeding a robust automatic determination of the optimal number of clusters. Given the noisy nature of biological data, robust measures like the Silhouette Width are preferable to noise-sensitive measures like the Dunn index, which is instable against outliers due to the consideration of only two distances. An inappropriate choice of method for cluster center determination might have been one of the reasons for the insufficient clustering results obtained by this validity measure.

The approaches described in this section are available as part of the Machaon CVE (Clustering and Validation Environment) [9]. This software platform has been designed to support clustering-based analyses of expression patterns including several data- and knowledge-driven cluster validity indices. The program and additional information may be found at http://www.cs.tcd.ie/ Nadia.Bolshakova/GOtool.html.

Furthermore, to determine the optimal number of clusters to be used in clustering data that contains some labeled samples, the authors in [47] present another measure of cluster structure compatibility with a given label assignment. The intuition is simple: on the one hand, clusters should be uniformly labeled and therefore penalize pairs of samples that are within the same cluster but have different labels. On the other hand, it is not acceptable to create unnecessary partitions and therefore penalize pairs of samples that have the same label, but are not within the same cluster. Formally, *the compatibility score* of a cluster structure with the training set is defined as the sum of two terms. The first is the number of tissue pairs $(\mathbf{v},\mathbf{u})$ such that $\mathbf{v}$ and $\mathbf{u}$ have the same label and are assigned to the same cluster. The second term is the number of $(\mathbf{v},\mathbf{u})$ pairs that have different labels and are assigned to different clusters. This score is also called *the matching coefficient* in the literature [2]. To handle label assignments defined only on a subset of the data, the comparison is restricted to count pairs of examples for which labels are assigned (the matching coefficient for a submatrix is computed). Based on this notion, using a binary search, the choice of clustering parameters can be optimized to find the most compatible clustering. It is also emphasized that this general idea can be applied to any parameter dependent clustering method and is not restricted to a particular choice.

3.7 A Normalization Technique for Cluster Validity Indices

The combined application of different intercluster/intracluster distances and different distances between two samples may produce validation indices of different scale ranges. Hence, the indices with higher values may have a stronger effect on the calculation of the average index values. This may result in a biased prediction of the optimal number of clusters. To overcome this problem the following normalization technique has been applied. Given a cluster configuration consisting of c clusters, for any partition $U_c : X \leftrightarrow X_1 \cup \ldots \cup X_c$, the normalized Dunn's indices $\mathbf{D}_{ij}^{*}$ (vectors) are calculated as:

$$D_{ij}^{*}\left(U_c\right) = \left(D_{ij}\left(U_c\right) - \overline{D}_{ij}\right) \Big/ \sigma D_{ij}, \tag{3.24}$$

$$\overline{D}_{ij} = \frac{1}{n}\sum_{k} D_{ij}\left(U_k\right), \tag{3.25}$$

where i reflects the selection of the intercluster distance calculation method $\left(i = 1,\ldots,6\right)$, j is the selection of the intracluster distance calculation method $\left(j = 1,\ldots,3\right)$, $D_{ij}\left(U_c\right)$ is the value of a Dunn's validity index, n is the number of partitions, σD_{ij} is the standard deviation of $D_{ij}\left(U_c\right)$ across all values of c. The normalized values of the eighteen Dunn's validity indices and their average indices at each number of clusters, c, for $c = 2$ to $c = 6$ are shown in Table 2. An examination of these results indicates that $c = 2$ represents the most appropriate partition for the data under analysis.

Furthermore, as far as the other validity indices are concerned, as we have already mentioned, different approaches of the used validity indices have been implemented. These appraches depend on the choice of the distance metric (Euclidean, Manhattan or Chebychev), and as far as C-index is concerned, its approaches also depend on the choice of the minimal C-index (two approaches implemented, discussed in Section 3.6). It has been observed that all these approaches may produce validation indices of different scale ranges. Hence, the indices with higher values may have a stronger effect on the calculation of the average index values. This may result in a biased prediction of the optimal number of clusters. To overcome this problem the above normalization technique used for Dunn index, is applied to all used validity measures, too. Thus, when-

ever it is necessary, the normalized Silhouette, C-index and Goodman-Kruskal indices may be calculated by equation (3.24) using the Silhouette, C-index and Goodman-Kruskal indices respectively instead of the Dunn's index.

Validity Index	c=2	c=3	c=4	c=5	c=6
$\mathbf{D_{11}}$	1.17	0.37	-1.50	0.32	-0.36
$\mathbf{D_{21}}$	1.71	-0.07	-0.22	-0.64	-0.78
$\mathbf{D_{31}}$	1.70	0.03	-0.30	-0.67	-0.76
$\mathbf{D_{41}}$	1.62	0.17	-0.23	-0.59	-0.97
$\mathbf{D_{51}}$	1.70	0.05	-0.34	-0.76	-0.65
$\mathbf{D_{61}}$	1.77	-0.57	-0.21	-0.59	-0.40
$\mathbf{D_{12}}$	1.37	0.46	-1.18	0.05	-0.71
$\mathbf{D_{22}}$	1.69	-0.02	-0.19	-0.64	-0.84
$\mathbf{D_{32}}$	1.66	0.11	-0.24	-0.66	-0.86
$\mathbf{D_{42}}$	1.60	0.20	-0.20	-0.60	-1.00
$\mathbf{D_{52}}$	1.66	0.12	-0.27	-0.73	-0.78
$\mathbf{D_{62}}$	1.76	-0.42	-0.17	-0.60	-0.57
$\mathbf{D_{13}}$	1.25	0.20	-1.50	0.31	-0.27
$\mathbf{D_{23}}$	1.72	-0.15	-0.17	-0.65	-0.75
$\mathbf{D_{33}}$	1.72	-0.08	-0.24	-0.69	-0.71
$\mathbf{D_{43}}$	1.65	0.08	-0.18	-0.61	-0.94
$\mathbf{D_{53}}$	1.72	-0.07	-0.28	-0.78	-0.60
$\mathbf{D_{63}}$	1.75	-0.64	-0.16	-0.60	-0.35
Average	1.62	-0.01	-0.42	-0.51	-0.68

Table 2 : Normalized Dunn's values using 3 types of intracluster measures and 6 types of intercluster measures [16].

3.8 A Weighted Voting Technique for Cluster Validity Indices

Another approach to estimate the optimal partition consists of the implementation of an aggregation method based on a weighted voting strategy. An example is shown in Table 3, based on the Dunn's indices from Table 2 by replacing the index values by weighted votes, whose values range from 1 to 5. Thus, for example, $\mathbf{D}_{11}$ represents the highest index value and suggests the partition $c = 2$ as the optimal partition, hence its weighted vote is equal to 5. On the other hand, $\mathbf{D}_{11}$ represents the smallest index value for partition $c = 4$, hence its weighted vote is equal

to 1. The average weighted vote for each cluster partition confirms that $c = 2$ represents the most appropriate prediction. This weighted voting strategy is applied to any validity index used in this thesis (i.e. Dunn, Silhouette, C-index and Goodman-Kruskal index).

Validity Index	**c=2**	**c=3**	**c=4**	**c=5**	**c=6**
$\mathbf{D_{11}}$	5	4	1	3	2
$\mathbf{D_{21}}$	5	4	3	2	1
$\mathbf{D_{31}}$	5	4	3	2	1
$\mathbf{D_{41}}$	5	4	3	2	1
$\mathbf{D_{51}}$	5	4	3	1	2
$\mathbf{D_{61}}$	5	2	4	1	3
$\mathbf{D_{12}}$	5	4	1	3	2
$\mathbf{D_{22}}$	5	4	3	2	1
$\mathbf{D_{32}}$	5	4	3	2	1
$\mathbf{D_{42}}$	5	4	3	2	1
$\mathbf{D_{52}}$	5	4	3	2	1
$\mathbf{D_{62}}$	5	3	4	1	2
$\mathbf{D_{13}}$	5	3	1	4	2
$\mathbf{D_{23}}$	5	4	3	2	1
$\mathbf{D_{33}}$	5	4	3	2	1
$\mathbf{D_{43}}$	5	4	3	2	1
$\mathbf{D_{53}}$	5	4	3	1	2
$\mathbf{D_{63}}$	5	1	4	2	3
Average	5.00	3.61	2.83	2.00	1.56

Table 3 : Predicting the correct number of clusters by weighted voting technique. The entries represent vote values based on Dunn's validation index using 3 types of intracluster measures and 6 types of intercluster measures [16].

3.9 Combination of Cluster Validity Indices

The above weighted voting technique (Section 3.8) may also be applied to fuse the results originating from different validation methods. Table 4, depicts the global silhouette values, **GSu** , for each partition, and the silhouette values, **S**, for each number of clusters, c, for $c = 2$ to $c = 6$. In this case $c = 2$ is suggested as the best clustering configuration for the examined data set. So, an example of using combination of various validity indices in order to estimate the cor-

rect number of clusters in a data set, is depicted in Table 5 for three validation techniques. This table was obtained from Table 3 and Table 4 by calculating the average weighted vote for each technique. Thus, after computing all validity indices, the average weighted vote for each cluster partition has been calculated, and $c = 2$ is suggested as the optimal partition. The applied validation techniques confirm that the partition consisting of two clusters represents the most appropriate representation for the data set under consideration.

c	GS_u	S_1	S_2	S_3	S_4	S_5	S_6
2	0.31	0.42	0.16				
3	0.25	0.25	0.13	0.36			
4	0.26	0.18	0.23	0.38	0.23		
5	0.29	0.31	0.21	0.37	0.22	0.27	
6	0.19	0.22	0.60	0.01	0.56	0.14	0.33

Table 4 : Global silhouette values for each partition, $\mathbf{GSu}$, and the silhouette values, $\mathbf{S}$, for each cluster defining a partition [16].

Validation Technique	c=2	c =3	c=4	c=5	c=6
Silhouette	5.00	2.00	3.00	4.00	1.00
Dunn's	5.00	3.61	2.83	2.00	1.56
Average	5.00	2.81	2.92	3.00	1.28

Table 5 : Predicting the correct number of clusters for medulloblastomas data by aggregation of multiple validation methods [16].

Finally, it is important to note that the above results in Tables 2-5 were obtained when $d(\mathbf{x},\mathbf{y})$ was calculated using the well-known Euclidean distance between samples. Table 6, summarizes the effects of three measures, $d(\mathbf{x},\mathbf{y})$ described before in Section 3.6 (i.e. Euclidean, Manhattan and Chebychev distance metrics). It suggests that the estimation of the optimal

partition by normalized and non-normalized indices is not sensitive to the implemented type of metrics, $d(\mathbf{x},\mathbf{y})$.

Validity Index **Based on Distances**	**c=2**	**c=3**	**c=4**	**c=5**	**c=6**
Euclidean	**0.93 (1.47)**	0.48 (-0.46)	0.45 (-0.08)	0.39 (-0.55)	0.40 (-0.38)
Manhattan	**1.70 (1.63)**	0.86 (-0.42)	0.79 (-0.09)	0.65 (-0.73)	0.66 (-0.40)
Chebychev	**0.90 (1.29)**	0.48 (0.10)	0.49 (-0.20)	0.39 (-0.61)	0.40 (-0.58)

Table 6 : Dunn's validity indices for expression clusters originating from leukaemia data. The entries represent the average Dunn's values based on the distances shown in Table 2 and using three measures for $d(\mathbf{x},\mathbf{y})$. Normalized Dunn's validity indexes are given between brackets. Bold entries represent the optimal number of clusters, c, predicted by each method [19].

3.10 Implementation of Cluster Validity Indices

In this study, we apply the validity indices presented in Section 3.6 using knowledge-driven methods (GO-based Wu and Palmer similarity measure) to estimate the number and the quality of the clusters. These validity indices could be used to support the discovery of clusters of genes sharing similar functions. Such clusters may indicate regulatory pathways, which could be significantly relevant to specific phenotypes or physiological conditions. Also, we apply the same validity indices using data-driven methods (Euclidean, Manhattan and Chebychev metrics) to estimate the number and the quality of the clusters, too. The normalization of index values (mentioned in Section 3.7) and the weighted voting strategy (mentioned Section 3.8) have also been implemented to improve the prediction procedure. We examine the comparison and combination of different data- and knowledge-driven cluster validity indices.

To sum up, several clustering techniques have been proposed to support the analysis of gene expression data. Determining the appropriate number of clusters in experimental data is a complex and time-consuming process. Cluster validity indices represent useful tools to guide unsupervised data analysis. They are particularly relevant for the estimation of clustering partitions in different applications, which may require the definition of the number of clusters beforehand. The combination of these methods may be used for cluster evaluation tasks. It has been shown

how these methods may support the prediction of the optimal cluster partition. The results also suggest that the normalization of index values and a weighted voting strategy may improve the prediction procedure. The normalization scheme may represent a more robust mechanism to predict the correct number of clusters. It allows smoothing the effect of the highest values on the calculation of the average index values. Moreover, it highlights subtle differences between index values originating from different clustering configurations. The advantage of a weighted voting approach lies in an aggregation of multiple validation methods in order to improve the estimation of the most adequate clustering partition for interpretation purposes. A systematic validation approach may significantly support genome expression analyses for knowledge discovery applications.

Finally, Figures 20-23 present in detail the data- and knowledge-driven cluster validity assessment system implemented in this thesis. Figure 23 shows the data- and knowledge-driven cluster validity assessment system implemented, while Figure 24, Figure 25 and Figure 26 present how Silhouette index or Goodman-Kruskal index, C-index and Dunn index work in detail at stage A in Figure 23.

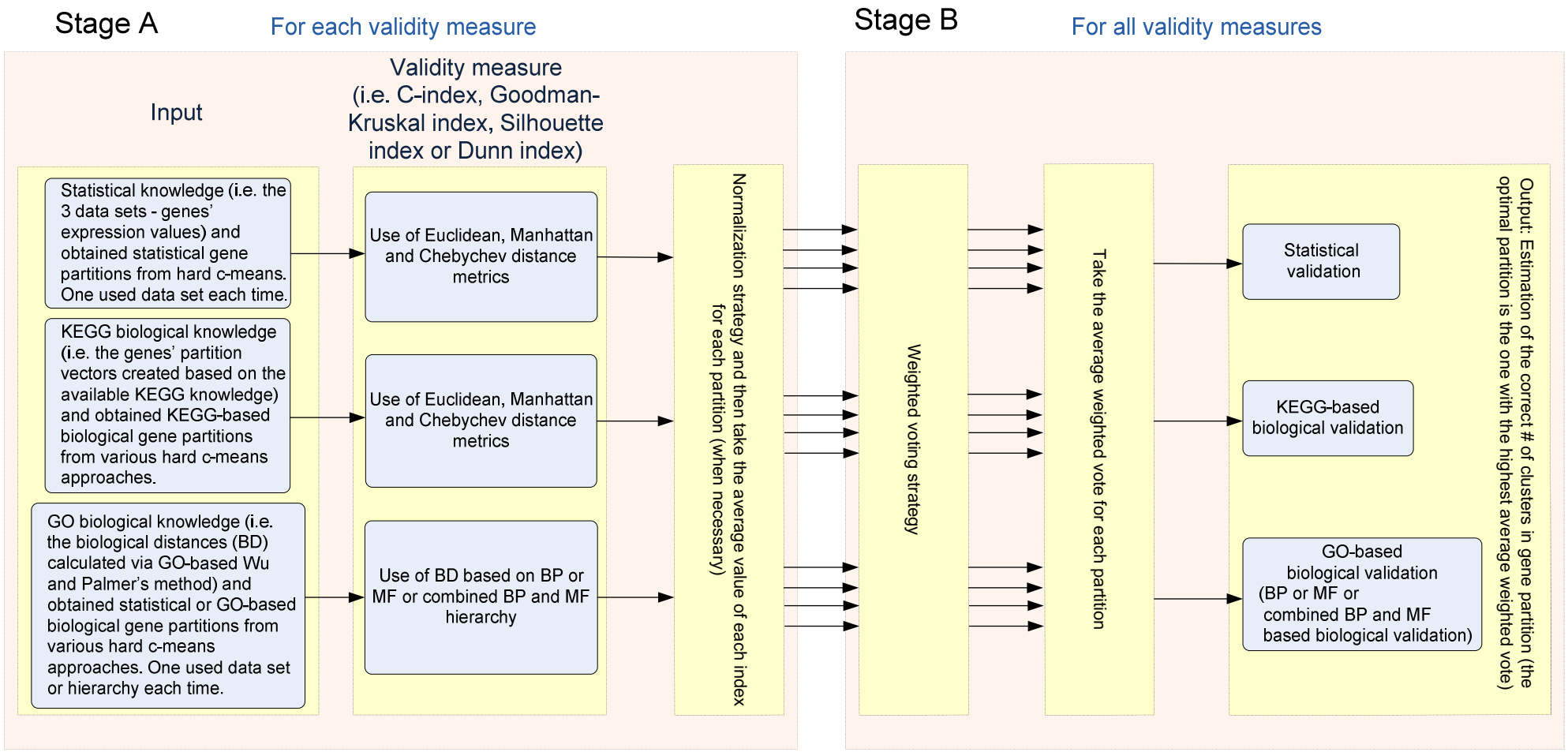


Figure 23 : The implemented data- and knowledge-driven cluster validity assessment system, presented as red bidirectional arrows in Figure 4.

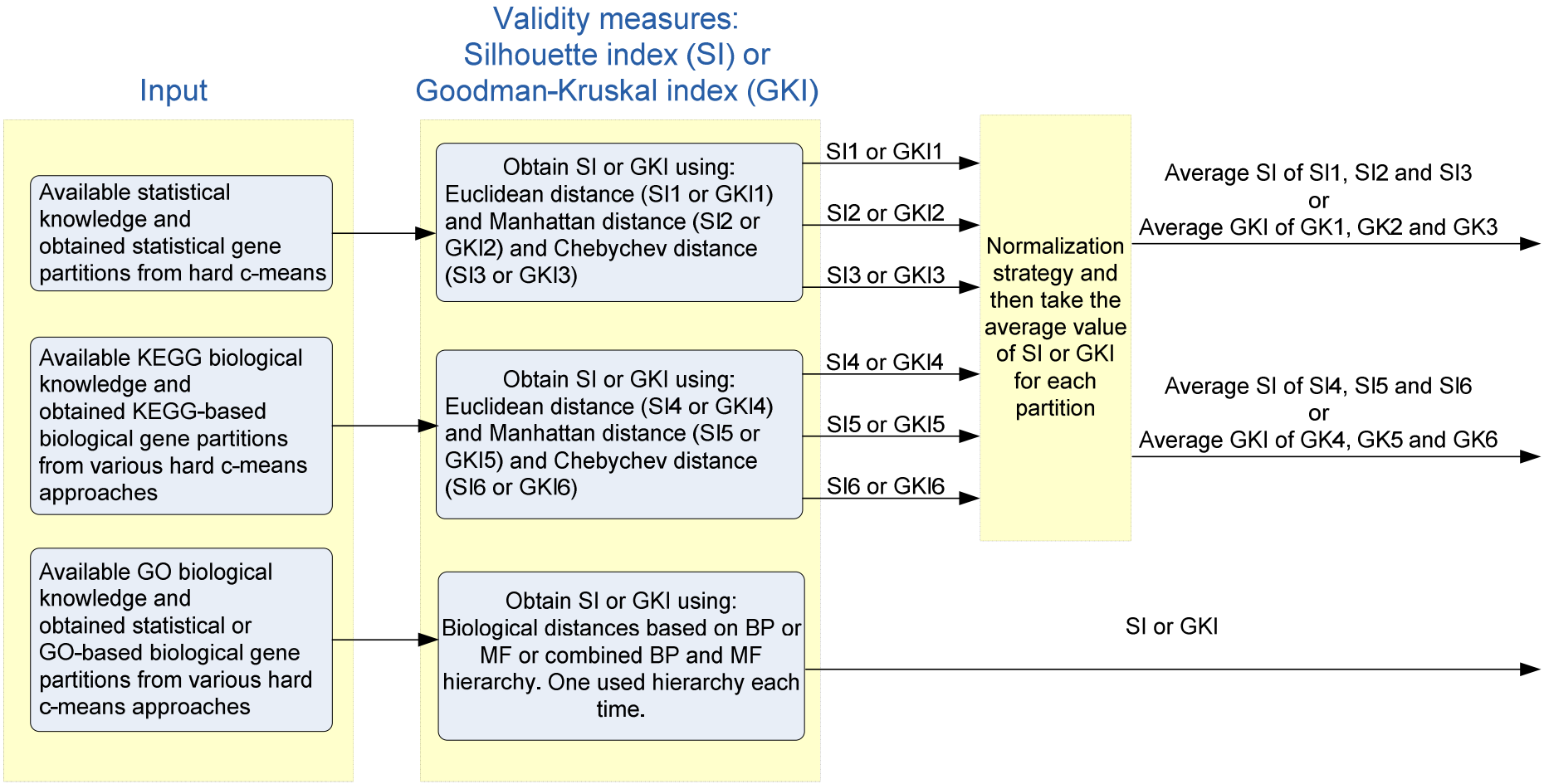


Figure 24 : How Silhouette index or Goodman-Kruskal index works in detail at stage A in Figure 23.

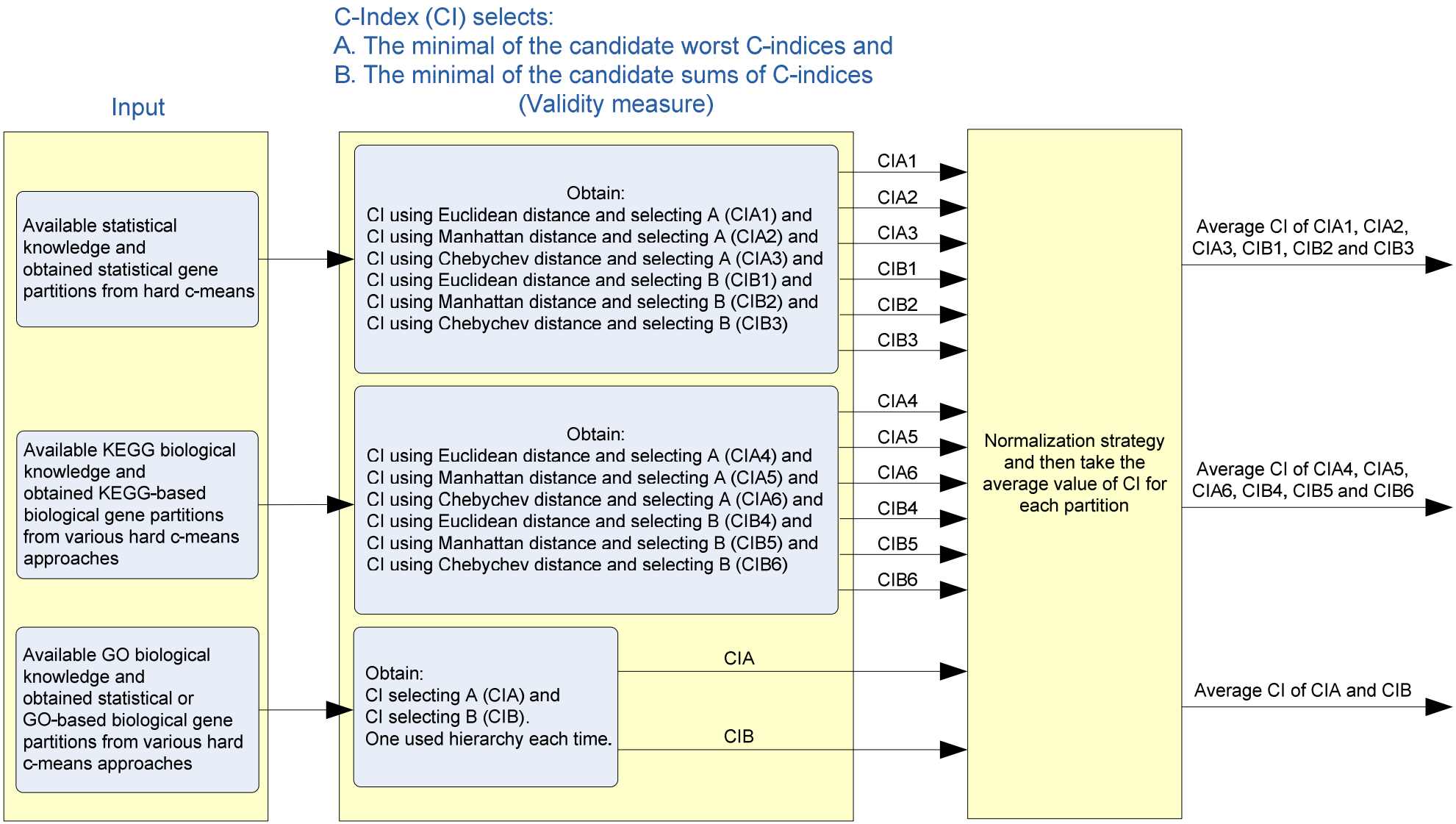


Figure 25 : How C-index works in detail at stage A in Figure 23.

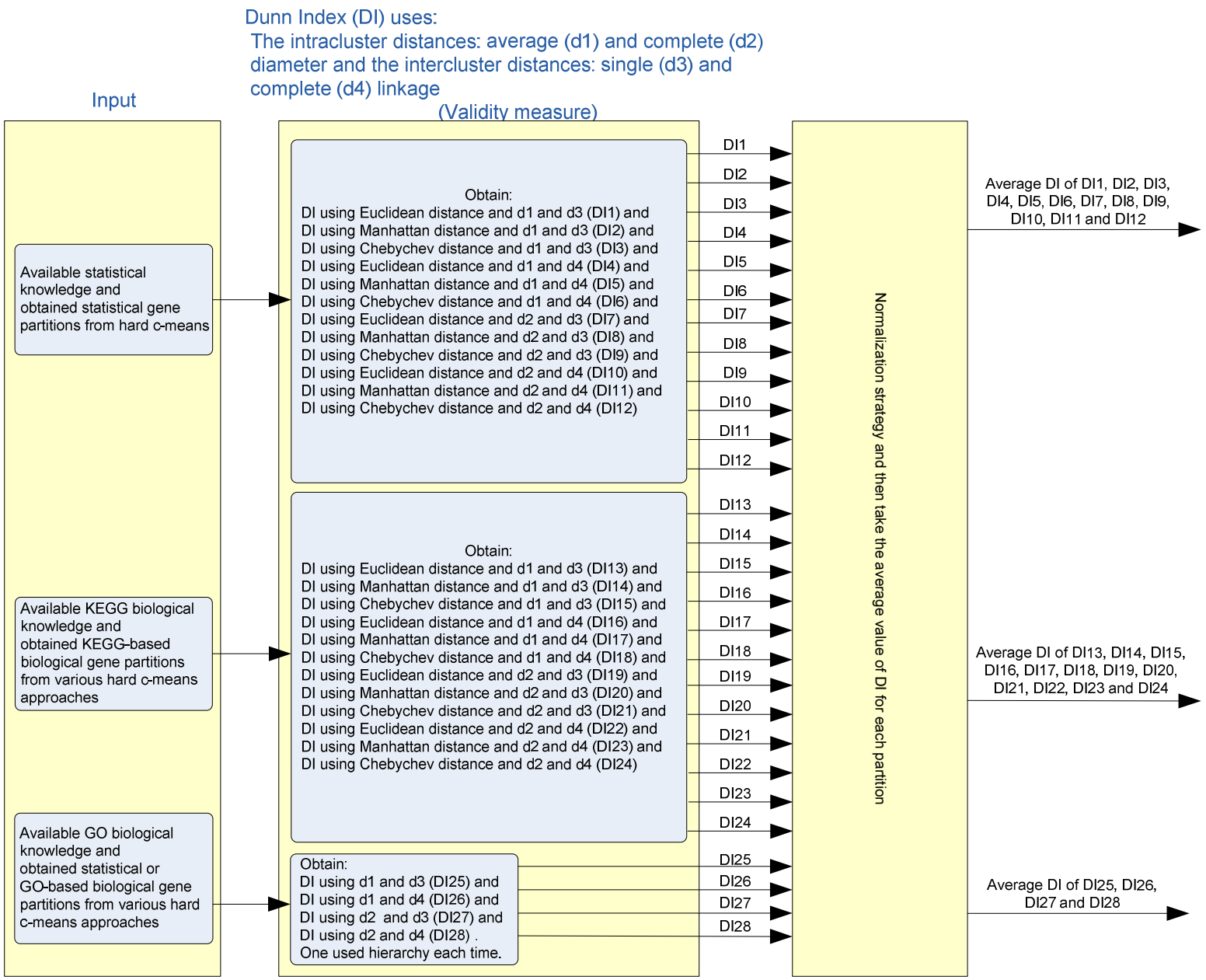


Figure 26 : How Dunn index works in detail at stage A in Figure 23.

3.11 Similarity Indices

The problem of measuring the correspondence between two partitions of an object set has attracted substantial interest in the literature of classification. One may be interested in assessing degree of similarity (or verifying equivalence) of two clustering algorithms (for example one being a simpler and/or more efficient version of the other). This is an important issue with current research, where large data sets are so common. Similarity indices can be used to compare partitions (clusterings) of a data set. Many such indices were introduced in the literature over the years. Indicatively, Table 7 shows some useful similarity indices. For further information the reader is encouraged to refer to [45]. Even though their values differ for the same clusterings that they compare, after correcting for agreement attributed to chance only, their values become similar and some of them even become equivalent. Consequently, the problem of choice of the index to be used for comparing different clusterings becomes less important.

We begin by reviewing a well-known measure of partition correspondence often attributed to the author in [39], the Rand index (R). The Rand index appears to be one of the most popular alternatives for comparing partitions and has a rather interesting history of being rediscovered and/or modified by different authors. Given an n object set $S=\{O_1,\ldots,O_n\}$, suppose $U=\{u_1,\ldots,u_R\}$ and $V=\{v_1,\ldots,v_c\}$ represent two different partitions of S, i.e. the entries in U and V are subsets of S, $\bigcup_{i=1}^{R}u_i=S=\bigcup_{j=1}^{C}v_j$, $u_i\cap u_i^{'}=\varnothing=v_j\cap v_j^{'}$, for $1\le i\ne i^{'}\le R$ and $1\le j\ne j^{'}\le C$. Letting n_{ij} denotes the number of objects that are common to classes u_i and v_j, the information on class overlap between the two partitions U and V can be written in the form of a contingency table (using standard "dot" notation for row and column sums) with $n_{i\cdot}$ and $n_{\cdot j}$ referring respectively to the number of objects in classes u_i (row i) and v_j (column j), as in Table 8.

No.	Introduced by	Symbol
1	Sokal and Michener (1958), Rand(1971)	R
2	Hamann (1961), Hubert (1977)	H
3	Czekanowski (1932), Dice (1945), Gower and Legendre (1986)	CZ
4	Kulczynski (1927)	K
5	McConnaughey (1964)	MC
6	Peirce (1884)	PE
7	Fowlkes and Mallows (1983), Ochiai (1957)	FM
8	Wallace (1) (1983)	W1
9	Wallace (2) (1983)	W2
10	Russel and Rao (1940)	RR
11	Goodman and Kruskal (1954), Yule (1927)	GK

Table 7 : Similarity Indices – References and Symbols.

		Partition V				
	Class	$\mathbf{v_1}$	$\mathbf{v_2}$	…	$\mathbf{v_C}$	**Sums**
Partition U	$\mathbf{u_1}$	n_{11}	n_{12}	…	n_{1C}	$n_{1\cdot}$
	$\mathbf{u_2}$	n_{21}	n_{22}		n_{2C}	$n_{2\cdot}$
	.	.	.		.	.
	.	.	.		.	.
	.	.	.		.	.
	$\mathbf{u_R}$	n_{R1}	n_{R2}	…	n_{RC}	$n_{R\cdot}$
	Sums	$n_{\cdot 1}$	$n_{\cdot 2}$	…	$n_{\cdot C}$	$n_{\cdot\cdot} = n$

Table 8 : Notation for Comparing Two Partitions.

The author in [39], as well as others, bases measures of correspondence between U and V on how object pairs are classified in the RxC contingency table. Specifically, there are four different types among the $\binom{n}{2}$ distinct pairs that could be found:

- type (i): objects in the pair are placed in the same class in U and in the same class in V
- type (ii): objects in the pair are placed in different classes in U and in different classes in V
- type (iii): objects in the pair are placed in different classes in U and in the same class in V
- type (iv): objects in the pair are placed in the same class in U and in different classes in V

Types (i) and (ii) are typically interpreted as agreements in the classification of the objects from a pair. Types (iii) and (iv) represent disagreements. Obviously, if A represents the total number of agreements and D the total number of disagreements, then $A+D=\binom{n}{2}$. Moreover, we can show [40] that

$$A=\binom{n}{2}+\sum_{i=1}^{R}\sum_{j=1}^{C}n_{ij}^2-\frac{1}{2}\cdot\left(\sum_{i=1}^{R}n_{i\cdot}^2+\sum_{j=1}^{C}n_{\cdot j}^2\right)=\binom{n}{2}+2\cdot\sum_{i=1}^{R}\sum_{j=1}^{C}\binom{n_{ij}}{2}-\left(\sum_{i=1}^{R}\binom{n_{i\cdot}}{2}+\sum_{j=1}^{C}\binom{n\cdot j}{2}\right), \quad (3.26)$$

where a binomial coefficient $\binom{m}{2}$ is defined as 0 when $m=0$ or 1. In fact, as given in Table 9, explicit formulae can be obtained to express the number of object pairs of each type as a function of n, $n_{i\cdot}$, $n_{\cdot j}$ and n_{ij}. If we assume that the marginal sums are fixed in the RxC contingency table, then all of the formulae in Table 9, including those given for the sums A and D, are constant linear transformations of $\sum_{i,j}n_{ij}^2$ and thus, of each other.

Type	Formula
(i)	$\frac{1}{2}\cdot\sum_{i=1}^{R}\sum_{j=1}^{C} n_{ij}\cdot\left(n_{ij}-1\right)$
(ii)	$\frac{1}{2}\cdot\left(n^2+\sum_{i=1}^{R}\sum_{j=1}^{C} n_{ij}^2-\left(\sum_{i=1}^{R} n_{i\cdot}^2+\sum_{j=1}^{C} n_{\cdot j}^2\right)\right)$
(iii)	$\frac{1}{2}\cdot\left(\sum_{j=1}^{C} n_{\cdot j}^2-\sum_{i=1}^{R}\sum_{j=1}^{C} n_{ij}^2\right)$
(iv)	$\frac{1}{2}\cdot\left(\sum_{i=1}^{R} n_{i\cdot}^2-\sum_{i=1}^{R}\sum_{j=1}^{C} n_{ij}^2\right)$
(i) + (ii) = A =	$\binom{n}{2}+\sum_{i=1}^{R}\sum_{j=1}^{C} n_{ij}^2-\frac{1}{2}\cdot\left(\sum_{i=1}^{R} n_{i\cdot}^2+\sum_{j=1}^{C} n_{\cdot j}^2\right)$
(iii) + (iv) = D =	$\frac{1}{2}\cdot\left(\sum_{i=1}^{R} n_{i\cdot}^2+\sum_{j=1}^{C} n_{\cdot j}^2\right)-\sum_{i=1}^{R}\sum_{j=1}^{C} n_{ij}^2$

Table 9 : Formulae for the Number of (Unordered) Object Pairs of the Four Types [44].

Intuitively, two partitions that are similar produce relatively large values of A and small values of D. Thus, depending on how A and D are normalized, different raw measures of agreement are possible, e.g. the author in [39] uses the Rand similarity index (R): $R = A\Big/\binom{n}{2}$, the authors in [42], [43] adopt $D\Big/\binom{n}{2}$ and the author in [41] suggests the Hubert similarity index (H): $H = \left(A-D\right)\Big/\binom{n}{2}$. In all these cases, the raw measures have straightforward probabilistic interpretations with respect to picking a pair of objects at random, i.e., $R = A\Big/\binom{n}{2}$ (i.e. the Rand si-

milarity index) is the probability of agreement, $D\Big/\binom{n}{2}$ is the (complementary) probability of a disagreement and $H=(A-D)\Big/\binom{n}{2}$ (i.e. the Hubert similarity index) is the difference between the probability of an agreement and a disagreement. From the above it follows that $0\le R\le 1$, $0\le D\Big/\binom{n}{2}\le 1$ and $-1\le H\le 1$ respectively.

With so many similarity indices available, some of them shown in Table 7, the choice of the index and subsequent interpretation of its value is not obvious. As an example, in Table 10 we conclude mean values and the upper and lower bound of the values for six selected similarity indices from Table 7 (FM, R, H, RR, CZ and W). The number of clusters requested was the same for both clusterings and equal 2, 3 and then 6 clusters. Clusterings were obtained at random and independently so the differences affecting each index must be caused by the agreement due to chance in a different way, that depends on its index formula. To eliminate the effect of agreement due to chance, a correction for the Rand (R) similarity index has been suggested. Any simi-

Index	**FM**	**R**	**H**	**RR**	**CZ**	**W**	**# Clusters**
Mean	0.678	0.499	-0.001	0.462	0.645	0.926	2
L	0.511	0.499	-0.002	0.261	0.511	0.524	
U	0.705	0.500	0.000	0.497	0.665	0.996	
Mean	0.494	0.417	-0.167	0.248	0.453	0.748	3
L	0.389	0.335	-0.331	0.151	0.384	0.455	
U	0.573	0.515	0.030	0.328	0.497	0.988	
Mean	0.265	0.547	0.094	0.071	0.236	0.430	6
L	0.208	0.374	-0.252	0.044	0.202	0.265	
U	0.335	0.656	0.312	0.113	0.265	0.683	

Table 10 : Six selected similarity indices [45].

larity index SI after such correction has a form

$$CSI = \frac{SI - E(SI)}{1 - E(SI)}, \tag{3.27}$$

where expectation $E(SI)$ is conditional upon fixed sets of marginal counts in the matrix N Table 8. Consequently the corrected value of the index should be close to 0 if the agreement is due to chance only and will be equal 1 when the uncorrected index equals 1.

Table 11 contains mean values and upper and lower bounds of the values for six similarity indices from Table 7 after they were corrected for chance agreement (CFM, CR, CH, CRR, CCZ and CW). Clusterings being compared were independent and there was no actual similarity. It can be seen that mean values in Table 11 are all either equal to zero or very close to zero. Additionally, results for indices CR, CH and CCZ are equal. As authors in [45] state that some of the indices become equivalent after correction for chance agreement (3.27) is applied. For example, Rand (R), Hubert (H), and Czekanowski (CZ) similarity indices are equivalent after correction for agreement due to chance.

Index	**CFM**	**CR**	**CH**	**CRR**	**CCZ**	**CW**	**# Clusters**
Mean	0.000	0.000	0.000	0.000	0.000	0.000	2
L	0.000	0.000	0.000	0.000	0.000	0.000	
U	0.002	0.002	0.002	0.001	0.002	0.003	
Mean	0.001	0.001	0.001	0.000	0.001	0.001	3
L	0.000	0.000	0.000	0.000	0.000	0.000	
U	0.007	0.006	0.006	0.002	0.006	0.004	
Mean	0.005	0.004	0.004	0.001	0.004	0.003	6
L	0.001	0.001	0.001	0.000	0.001	0.001	
U	0.011	0.010	0.010	0.002	0.010	0.007	

Table 11 : Six corrected similarity indices [45].

From all above, we can conclude that after such correction all similarity indices are either equal to zero or very close to zero. Even though their values differ for the same clusterings that they compare, after correcting for agreement attributed to chance only, their values become similar and some of them even become equivalent. Consequently, with so many similarity indices available, the problem of the choice of an appropriate similarity index to be used for comparing different clusterings, becomes less important.

We also note that similarity indices can be used to evaluate a single clustering procedure and also to compare two clustering methods (or two algorithms of the same method). Furthermore, the behavior of the similarity index can also be used as an indicator of the proper number of clusters in a data set. Interesting results can be found in [46].

3.12 Application in Multiple Data Sets

The gene clustering methodologies implemented in this thesis, are shown in detail in Figure 28. Two different types of biological knowledge are available: GO hierarchies, which are distinguished to BP, MF and combined BP-MF hierarchy, and KEGG PWs. It is noted that from GO hierarchies we compute the biological distances via Wu and Palmer's method. A statistical knowledge consisting of three data sets on breast cancer (Sotiriou's, Veer's and Sorlie's data set) is also available. To apply gene clustering methodologies in a single or multiple data sets a pre-processing is required. More information about this necessary pre-processing is given in Section 3.2. It is important to note that after pre-processing, the genes among the three data sets are common and the clustering procedures are applied independently to each data set.

The three data sets can be used after an appropriate pre-processing, as shown in Figure 28. Then, the hard c-means clustering method is applied, using Euclidean distance metric on gene expression values. We express this procedure as **statistical clustering**. Another approach of executing hard c-means clustering is based on the biological distances calculated before, using one GO hierarchy each time. We express this procedure as **GO biological clustering**.

In order to validate the statistical partition emerged from statistical clustering, a statistical and a GO biological cluster validation are required. The **statistical cluster validation** applies several validity methods (i.e. C-index, Silhouette index, Dunn index and Goodman-Kruskal index) that

use three types of distance metrics on gene expression values: Euclidean, Manhattan and Chebychev. Except the gene expression values, as shown in Figure 28, the cluster validation methods use also the calculated biological distances, using one GO hierarchy each time. This is expressed as **GO biological cluster validation**. To validate the biological partition emerged from GO biological clustering approach, a GO biological cluster validation is required. It is noted that a normalization and a weighted voting strategy are applied to improve the statistical or GO biological cluster validation. As the red arrows imply, the clustering method and the validity measures are executed multiple times with different input parameter values, i.e. number of clusters, until some candidate optimal statistical or biological partitions are obtained.

Another approach to obtain biological partitions, is via KEGG knowledge, that also requires some pre-processing, i.e. mapping genes to the Entrez Gene nomenclature and construction of genes' partition vectors. Then, the clusters are obtained using three methods, expressed as KEGG1, KEGG2 and KEGG3 biological clustering. The first method **(KEGG1 biological clustering)** is based on the idea that genes that take part in at least m common PWs must belong to the same cluster. After many trials we conclude that the best choice is $m=1$. As we can see from the Figure 27, when $m=1$ the obtained clusters are characterized more from both internal homogeneity and external separation than when $m>1$. From Table 12 we also see that when $m=1$ fewer clusters share the same information, i.e. more than 1 cluster have genes belong to common KEGG PWs, than when $m>1$. Also, when $m=1$ the obtained clusters that have at most 2 genes, are fewer than when $m>1$ and the mean number of the genes per cluster is 9, while when $m=2$ it is 4 genes per cluster. Thus, apparently $m=1$ is the best choice.

The second method **(KEGG2 biological clustering)** is the hard c-means clustering method, applied using Euclidean or Correlation distance metric on genes' partition vectors. The last method **(KEGG3 biological clustering)** is originated in this thesis. Its basic concept is that the gene with the most common PWs to a cluster's center, is assigned to that cluster. The biological clusters from the two last methods are validated via the validity measures defined earlier, i.e. C-index, Silhouette index, Dunn index and Goodman-Kruskal index. All validity measures use Euclidean, Manhattan and Chebychev distance metrics on genes' partition vectors. This is referred to as **KEGG biological cluster validation**. It is noted that a normalization and a weighted voting strategy are also applied to improve the KEGG biological cluster validation. As shown in

Figure 28, the two last methods require the number of clusters as an input parameter and are executed repeatedly until some candidate optimal partitions are obtained. Thus, from KEGG2 and KEGG3 biological clustering we obtain some candidate optimal biological partitions. Next, these partitions are compared with the one obtained from the first method (KEGG1 biological clustering). For this purpose several methods are available, i.e. Rand index, Hubert index and corrected Rand index. The final selected partition is the one that converges most to that of the first method and is characterized as optimal.

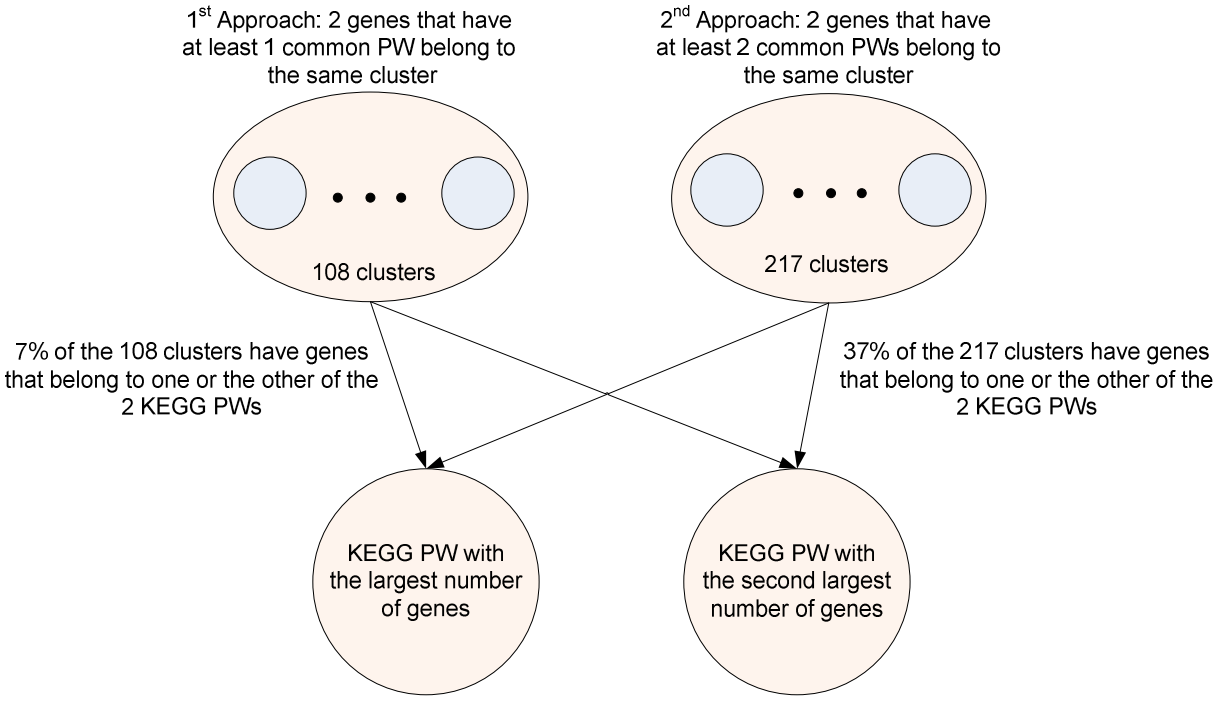


Figure 27 : An useful observation that justifies the choice $m=1$.

At Least Common PWs	# Obtained Clusters	% of Obtained Clusters that have at most 2 genes	Genes/Cluster
1	108	40	9
2	217	51	4

Table 12 : Useful statistics about the obtained partitions from KEGG knowledge using the 1st method.

In a similar way, we compare independently the biological partitions obtained from GO knowledge and the statistical partitions obtained from the statistical knowledge, i.e. the three data

sets, with the biological partition obtained from KEGG knowledge through the first method. Thus, we estimate the optimal biological and statistical partition that converges most to the biological partition obtained from KEGG knowledge through the first method.

It is noted that a more detailed explanation on the gene clustering methodologies results and the comparisons of all possible results is given in the next chapter, which deals with the results interpretation.

More details about the gene clustering methodologies implemented in this thesis are provided throughout the previous Sections.

3.13 Summary

In this chapter we present in detail all implemented gene clustering methodologies. The main objective of the research is to design clustering and cluster validity methods to estimate the number of clusters in gene expression datasets. C-means methodology is a commonly used clustering technique, which aims to partition n observations into k clusters in a way that each observation belongs to the cluster with the nearest mean. Various cluster validity indices have been proposed to measure the quality of clustering results. The Gene Ontology (GO) and Kyoto Encyclopedia of Genes and Genomes (KEGG) databases represent the up to date biological knowledge. A traditional node-counting method (i.e. the GO-based Wu and Palmer's method) has been implemented to measure knowledge-based similarity between genes products (biological distances), calculating the distance between the nodes associated with these terms in a hierarchy. To take advantage of the available biological information an enrichment cluster analysis using GO terms or KEGG pathways is also carried out. Furthermore, combination of GO-based (knowledge-driven) and microarray data (data-driven) validation methods may be used for the estimation of the number of clusters. A normalization and a weighted voting technique are usually used to improve the prediction of the number of clusters based on different data mining techniques. Also, a variety of similarity indices exist to compare partitions (clusterings) of a data set. Finally, the behavior of the similarity indices can also be used as an indicator of the proper number of clusters in a data set.

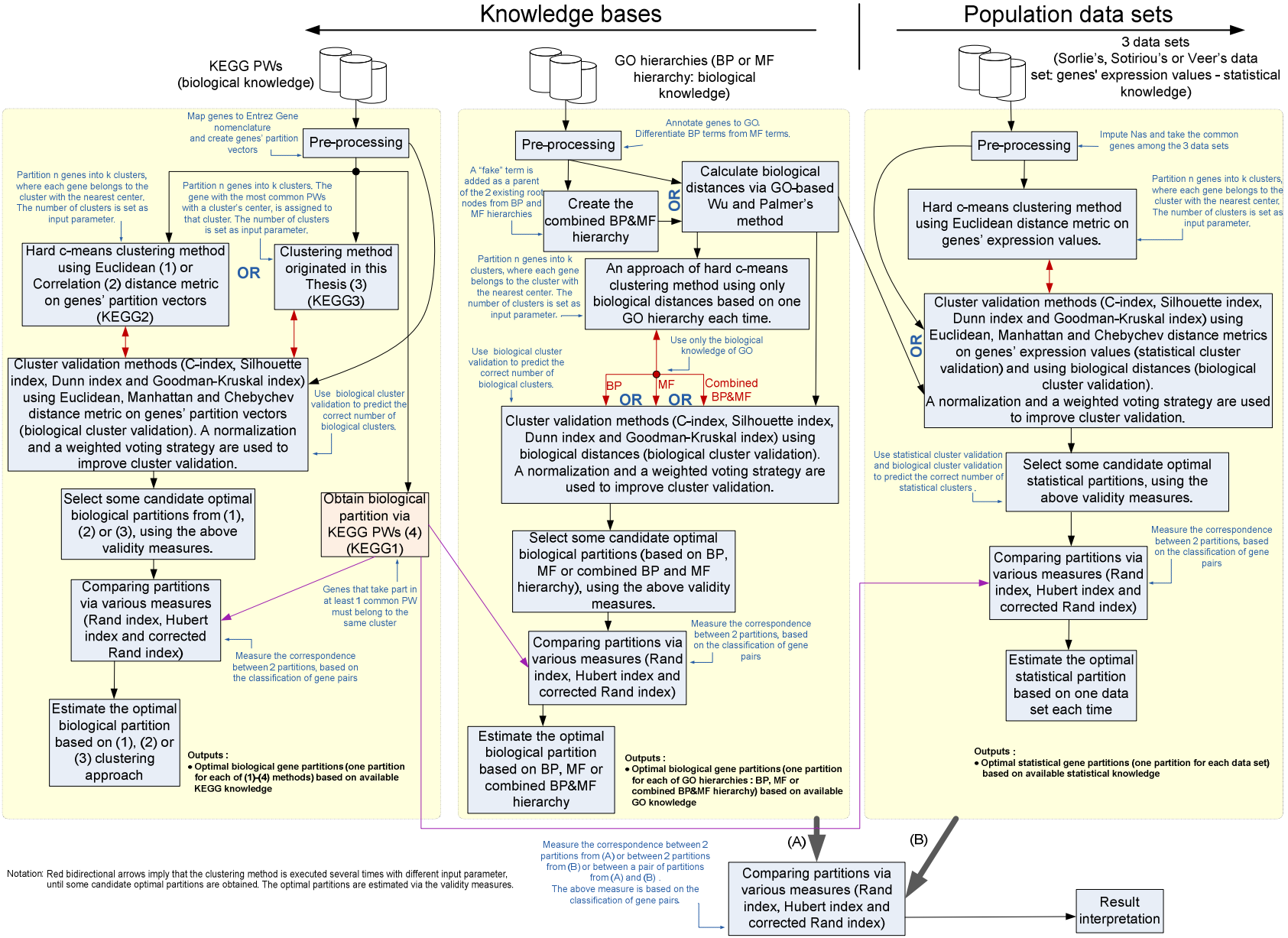


Figure 28 : The gene clustering methodologies implemented in this thesis in detail.

CHAPTER 4: RESULTS INTERPRETATION AND CONCLUSION

4.1 Introduction

In previous chapter the implemented gene clustering methodologies, shown in Figure 28 are analyzed in detail. In this chapter the results of these methodologies are presented and interpreted. A variety of comparisons between different obtained partitions take place, leading to meaningful biological conclusions. The purpose of this thesis, is to obtain gene clusters with biological meaning via various procedures, using the available statistical (i.e. the three data sets) or biological (i.e. the GO hierarchies) knowledge. The above gene clusters have biological meaning, if they converge to the gene clusters obtained by taking advantage of another type of biological knowledge, i.e. the available KEGG PWs knowledge. In this chapter we will present a comparative experimental evaluation of the implemented gene clustering methodologies, aiming at

illustrating their advantages and disadvantages. All the above methods are applied to the common genes among the three data sets that can be annotated to the GO and can be also mapped to the Entrez Gene nomenclature, i.e. finally to 946 distinct genes (see Section 3.2).

4.2 KEGG-Based Biological Gene Clustering

As stated in the previous chapter, an approach to obtain biological partitions is via KEGG knowledge after some pre-processing, i.e. mapping genes to the Entrez Gene nomenclature and the construction of the genes' partition vectors, all discussed in Section 3.2. Here we do not use none of the available data sets with genes' expression values. We take advantage only of the KEGG knowledge concerning the pathways each gene of the 946 genes takes part in. The clusters are obtained using three methods (*KEGG1, KEGG2 and KEGG3 biological clustering*). We have presented these methods in detail in Section 3.12. It is noted that the number of biological clusters obtained from the first method, i.e. KEGG1 biological clustering, are 108. More details about the first method have been discussed in Section 3.12. The biological clusters from the two last methods, i.e. KEGG2 and KEGG3 biological clustering, are validated via the validity measures, i.e. C-index, Silhouette index and Dunn index, using Euclidean, Manhattan and Chebychev distance metrics on genes' partition vectors (*KEGG biological cluster validation*). It is mentioned that although Goodman-Kruskal index has been implemented, due to its high computational complexity it has been excluded from the methodology to estimate the correct number of clusters. It is noted that a normalization and a weighted voting strategy are also applied to improve the KEGG biological cluster validation.

As shown in Figure 28, the two last methods require the number of clusters as an input parameter and are executed repeatedly until some candidate optimal partitions are obtained. Thus, from KEGG2 and KEGG3 biological clustering we obtain some candidate optimal biological partitions. Since a large number of clusters and very small clusters are not desired, we focus on a specific area of candidate optimal numbers of biological clusters, i.e. candidate optimal biological partitions, and we finally select the candidates with the highest votes. Then, Rand index (see Section 3.11), which is a similarity measure between KEGG2 or KEGG3 candidate biological partitions and KEGG1 biological partition, is used to estimate the optimal KEGG2 or KEGG3 biological partition. The results obtained are shown below.

From KEGG biological cluster validation concerning KEGG2 biological clustering which uses Correlation distance metric, the candidate numbers of biological clusters (biological partitions) with the highest votes are shown in Table 13. We finally select as the optimal number of biological clusters the number 130, because the hard c-means clustering method gives less empty clusters than when is executed with 100 biological clusters as input. Furthermore, Table 14 shows that the above partition with 130 clusters might be a good choice of optimal partition, since it corresponds to large value of Goodman-Kruskal index. Also, in Figure 29 the behavior of votes and Rand index as a function of a large range of partition values, concerning KEGG2 which uses Correlation distance metric is illustrated. The figure indicates that the partition with 130 clusters is a good choice as optimal partition of all candidates.

# Clusters	Votes
100	8.0000
130	8.0000

Table 13 : KEGG biological cluster validation concerning KEGG2 biological clustering which uses Correlation distance metric.

# Clusters	GK Index
20	-0.6829
50	0.0845
100	0.4548
150	0.8201
200	0.9902

Table 14 : Goodman-Kruskal (GK) index values for some partitions concerning KEGG2 which uses Correlation distance metric.

Table 15 shows the candidate numbers of biological clusters (biological partitions) with the highest votes. These results are obtained via KEGG biological cluster validation, concerning KEGG2 biological clustering which uses Euclidean distance metric. We finally select as the

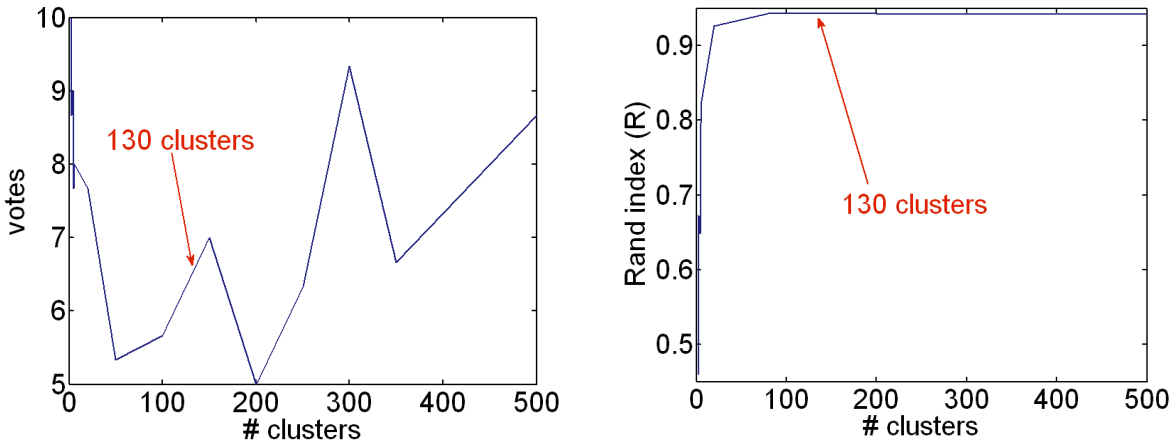


Figure 29 : Behavior of votes and Rand index for different partitions concerning KEGG2 which uses Correlation distance metric.

optimal number of biological clusters the number 110, because the hard c-means clustering method gives less empty clusters than the case with input parameter 80 biological clusters. Furthermore, from Table 16 we see that the above partition with 110 clusters could be considered as optimal partition, since it corresponds to large value of Goodman-Kruskal index. Also, Figure 30 shows the behavior of votes and Rand index as a function of a large range of partition values, concerning KEGG2 which uses Euclidean distance metric. Apparently, the partition with 110 clusters is a good choice as optimal partition of all candidates.

# Clusters	Votes
80	10.6667
110	7.6667

Table 15 : KEGG biological cluster validation concerning KEGG2 biological clustering which uses Euclidean distance metric.

In case of KEGG biological cluster validation concerning KEGG3 biological clustering, the candidate numbers of biological clusters (biological partitions) with the highest votes are shown in Table 17. Thus, the optimal number of biological clusters is 100. Furthermore, as Table 18 shows, the above partition with 100 clusters might considered as optimal, since it corresponds to large value of Goodman-Kruskal index. Also, the behavior of votes and Rand index as a function of a large range of partition values, concerning KEGG3, is shown in Figure 31 and indicates that

the partition with 100 clusters can be thought as optimal. The figure indicates that the partition with 100 clusters is a good choice as optimal partition of all candidates.

# Clusters	GK Index
20	-0.5837
50	-0.1159
100	0.5149
150	0.9357
200	0.9157

Table 16 : Goodman-Kruskal (GK) index values for some partitions concerning KEGG2 which uses Euclidean distance metric.

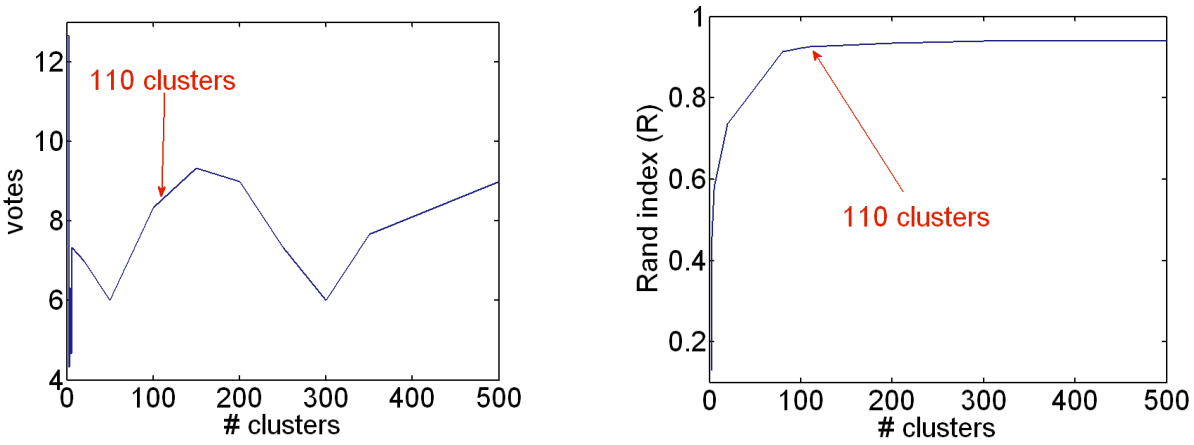


Figure 30 : Behavior of votes and Rand index for different partitions concerning KEGG2 which uses Euclidean distance metric.

# Clusters	Votes
100	8.0000

Table 17 : KEGG biological cluster validation concerning KEGG3 biological clustering.

From all the above figures we see that there is a variety of candidate optimal KEGG-based biological partitions with the highest votes. However, we select as optimal the partition which has a number of clusters close to that one of the obtained KEGG1 biological partition. We also

justify our choice with the behavior of Rand index, as shown in the above figures, where we observe that Rand index has achieved its highest value for this estimated optimal biological partition. That means that the above choice seems to be the optimal that converges mostly to KEGG1 biological partition.

# Clusters	GK Index
20	1.0962
50	0.6159
100	0.6064
150	-0.6194
200	-0.0323

Table 18 : Goodman-Kruskal (GK) index values for some partitions concerning KEGG3.

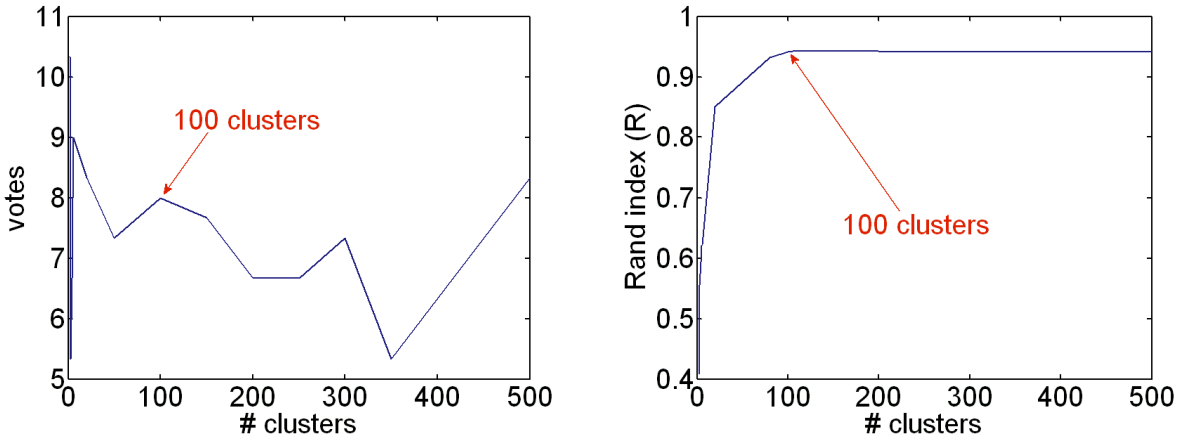


Figure 31 : Behavior of votes and Rand index for different partitions concerning KEGG3.

Next, the above optimal biological partitions, i.e. KEGG2 and KEGG3 biological partitions, are compared with the one obtained from the first method, i.e. with KEGG1 biological partition. For this purpose several similarity measures are available, i.e. Rand index, Hubert index and corrected Rand index. We finally conclude that the optimal KEGG2 and KEGG3 biological partitions converge strongly to the biological partition obtained from the first method, i.e. from KEGG1 biological clustering. This fact justifies our choice for the optimal KEGG2 and KEGG3 biological partitions, we made before. The results obtained are shown below. It is important to

note that as it has been already mentioned in Section 3.11, the Rand (R) and Hubert (H) similarity indices are equivalent after correction for agreement due to chance.

- KEGG1 biological partition compared with estimated optimal KEGG2 biological partition obtained using Correlation distance metric
  Rand index = 0.9430
  Hubert index = 0.8860
  Corrected Rand index = 0.2136

- KEGG1 biological partition compared with estimated optimal KEGG2 biological partition obtained using Euclidean distance metric
  Rand index = 0.9305
  Hubert index = 0.8609
  Corrected Rand index = 0.2551

- KEGG1 biological partition compared with estimated optimal KEGG3 biological partition
  Rand index = 0.9422
  Hubert index = 0.8843
  Corrected Rand index = 0.2391

Overall, the three methods (KEGG1, KEGG2 and KEGG3 biological clustering) obtain biological partitions very similar to the biological partition obtained from the first method (KEGG1 biological clustering). That means that we can obtain successfully the optimal biological partition from the available KEGG PWs knowledge via four different methods. The final selected biological partitions obtained from the three methods (i.e. KEGG2 biological clustering using Correlation distance metric, KEGG2 biological clustering using Euclidean distance metric and KEGG3 biological clustering) are those that converge most to the one obtained from the first method, and they are characterized as optimal. So, the obtained optimal biological partitions based on KEGG PWs knowledge are: the biological partition with 108 clusters obtained from the KEGG1 method, the biological partition with 130 clusters obtained from the KEGG2 biological clustering method using Correlation distance metric, the biological partition with 80 clusters obtained from the KEGG2 biological clustering method using Euclidean distance metric and the biological partition with 100 clusters obtained from the KEGG3 biological clustering method.

Figure 32 is based on our results and illustrates that genes are finally "well-clustered" via KEGG1, KEGG2 or KEGG3 method.

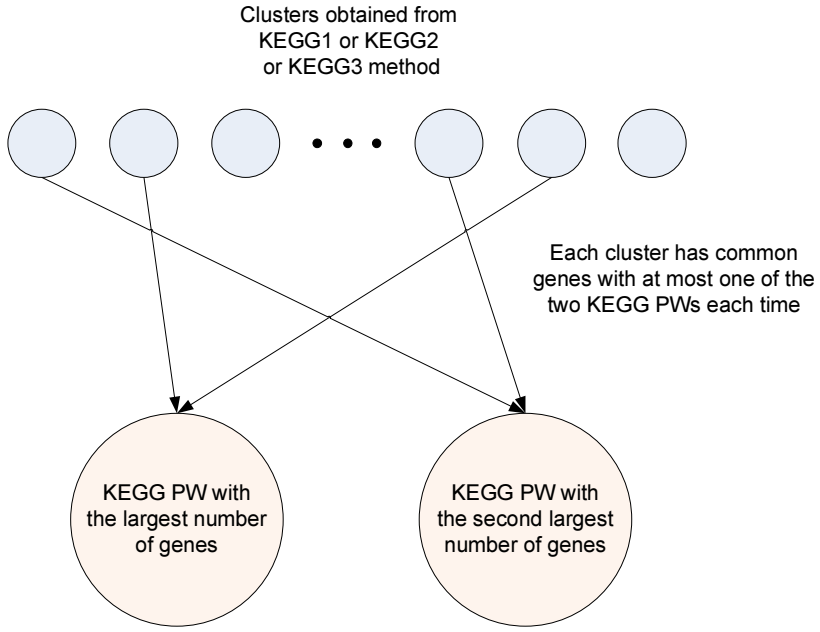


Figure 32 : An illustrative example that shows the obtained "clear-cut" clusters.

4.3 Data Sets –Based Statistical Gene Clustering

As Figure 28 shows, the three available data sets (Sorlie's, Sotiriou's and Veer's data set) should be pre-processed before they can be used. After pre-processing, we finally keep only the 14% of genes from Sorlie's data set, the 32% of genes from Sotiriou's data set and the 6% of genes from Veer's data set. This will not result in a biased results interpretation, since we finally keep the most important genes concerning breast cancer. This issue is further analyzed in Section 3.2.

The hard c-means clustering method is applied, using Euclidean distance metric on gene expression values, i.e. on each data set with the 946 genes independently (*statistical clustering*). To validate the statistical partition emerged from the above statistical clustering approach, a statistical and a GO biological cluster validation are required. Several validity methods (i.e. C-index, Silhouette index and Dunn index) are applied, using three types of distance metrics, i.e. Euclidean, Manhattan and Chebychev, on gene expression values (*statistical cluster validation*). It is mentioned that Goodman-Kruskal index has been implemented in this thesis, but due to its high

computational complexity it has not been applied it in our methodology to estimate the correct number of clusters. Except the gene expression values, as shown in Figure 28, the cluster validation methods use also the calculated GO-based biological distances (*GO biological cluster validation*), using one hierarchy (BP or MF or combined BP and MF hierarchy) each time. It is noted that a normalization and a weighted voting strategy are also applied to improve the statistical or GO biological cluster validation. As the red arrows imply, the clustering method and the validity measures are executed multiple times with different input parameter values, i.e. number of clusters, until some candidate optimal statistical partitions are obtained. Since a large number of clusters and very small clusters should be avoided, we focus on a specific area of candidate optimal numbers of statistical clusters, i.e. candidate optimal statistical partitions, and we finally select the candidates with the highest votes. Then, Rand index (see Section 3.11), which is a similarity measure between candidate statistical partitions and KEGG1 biological partition, is used to estimate the optimal statistical partitions. The results obtained are shown below, for the three available data sets.

4.3.1 Sorlie's Data Set -Based Statistical Gene Clustering

As far as Sorlie's data set is concerned:

- From statistical cluster validation concerning statistical clustering, the candidate numbers of statistical clusters (statistical partitions) with the highest votes are shown in Table 19.
- From BP biological cluster validation concerning statistical clustering, the candidate numbers of statistical clusters (statistical partitions) with the highest votes are shown in Table 20.
- From MF biological cluster validation concerning statistical clustering, the candidate numbers of statistical clusters (statistical partitions) with the highest votes are shown in Table 21.
- From combined BP and MF biological cluster validation concerning statistical clustering, the candidate numbers of statistical clusters (statistical partitions) with the highest votes are shown in Table 22.

It is noted that each partition does not contain empty clusters. Next, we compare independently the candidate optimal statistical partitions obtained from Sorlie's data set (shown in Tables 19-22), with the biological partition obtained from KEGG knowledge through the first method, i.e.

# Clusters	Votes
50	8.6667
60	9.3333
70	9.0000

Table 19 : Sorlie's data set statistical cluster validation concerning statistical clustering.

# Clusters	Votes
50	9.0000
70	8.6667
90	8.0000

Table 20 : Sorlie's data set BP biological cluster validation concerning statistical clustering.

# Clusters	Votes
50	10.3333
60	9.0000
70	8.0000

Table 21 : Sorlie's data set MF biological cluster validation concerning statistical clustering.

# Clusters	Votes
50	9.0000
60	9.3333
70	8.0000
90	7.6667

Table 22 : Sorlie's data set combined BP and MF biological cluster validation concerning statistical clustering.

through KEGG1 biological clustering. For this purpose we use the Rand index. Thus, we estimate the optimal statistical partition that converges most to the biological partition obtained from KEGG knowledge through the first method. The results obtained are shown in Table 23. From the results in Table 23, we estimate as the optimal Sorlie's data set-based statistical partition, the one with the maximum Rand index (R), i.e. the partition with 90 clusters. Furthermore, from Table 24 we see that the above partition with 90 clusters might be a good choice as optimal partition, since it corresponds to large value of Goodman-Kruskal index. We do not examine Goodman-Kruskal index using biological distances (biological cluster validation) even for few partitions, due to higher computational complexity than using statistical distances (statistical cluster validation), shown before. Also, in Figure 33 the behavior of votes and Rand index as a function of a large range of partition values, concerning Sorlie's data set is shown, indicating that the partition with 90 clusters is a good choice as optimal partition of all candidates. As far as all implemented similarity indices are concerned, for this estimated optimal statistical partition compared with the KEGG1 biological partition it holds:

Rand index = 0.9207

Hubert index = 0.8413

Corrected Rand index = 0.0032

So, the optimal Sorlie's data set-based statistical partition converges strongly to the biological partition obtained from the first method, i.e. from KEGG1 biological clustering. These results confirm our decision for the optimal Sorlie's data set-based statistical partition, made before.

# Clusters	R
50	0.9085
60	0.9112
70	0.9143
90	0.9207

Table 23 : Sorlie's data set-based candidate optimal statistical partitions.

# Clusters	GK Index
20	-1.4478
50	-0.4491
100	0.1618
150	0.6031
200	1.1319

Table 24 : Goodman-Kruskal (GK) index values for some partitions based on statistical cluster validation concerning statistical clustering on Sorlie's data set.

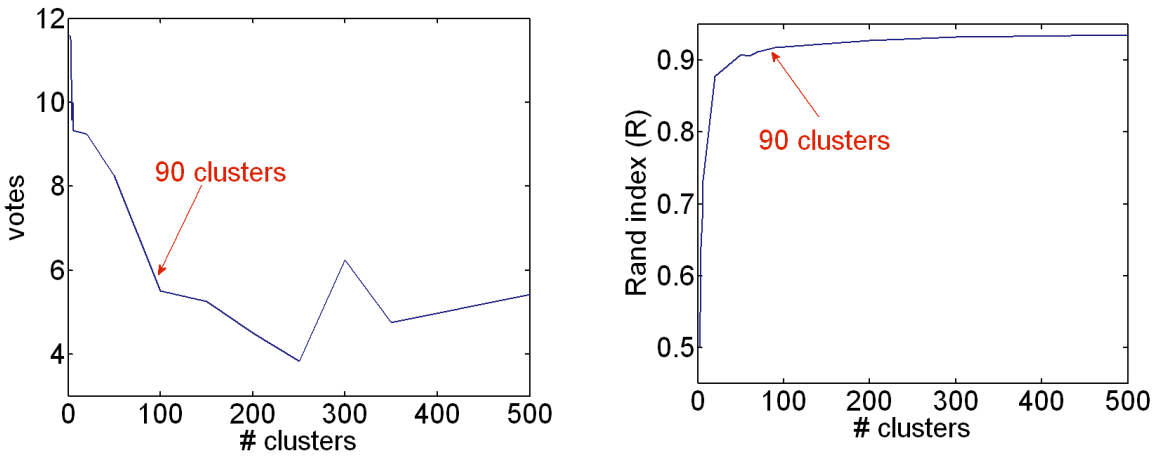


Figure 33 : Behavior of votes and Rand index for different partitions concerning Sorlie's data set.

4.3.2 Sotiriou's Data Set -Based Statistical Gene Clustering

As far as Sotiriou's data set is concerned:

- From statistical cluster validation concerning statistical clustering, the candidate numbers of statistical clusters (statistical partitions) with the highest votes are shown in Table 25.
- From BP biological cluster validation concerning statistical clustering, the candidate numbers of statistical clusters (statistical partitions) with the highest votes are shown in Table 26.
- From MF biological cluster validation concerning statistical clustering, the candidate numbers of statistical clusters (statistical partitions) with the highest votes are shown in Table 27.
- From combined BP and MF biological cluster validation concerning statistical clustering, the candidate numbers of statistical clusters (statistical partitions) with the highest votes are shown in Table 28.

It is noted that in each partition there are not obtained empty clusters. Next, we compare independently the candidate optimal statistical partitions obtained from Sotiriou's data set (shown in Tables 25-28), with the biological partition obtained from KEGG knowledge through the first method, i.e. through KEGG1 biological clustering. For this purpose we use Rand index. Thus, we estimate the optimal statistical partition that converges most to the biological partition obtained from KEGG knowledge through the first method. The results obtained are shown in Table 29. From the results in Table 29, we estimate as the optimal Sotiriou's data set-based statistical partition, the one with the maximum Rand index (R), i.e. the partition with 110 clusters. Furthermore, from Table 30 we see that the above partition with 110 clusters is a good choice as optimal partition, since it corresponds to large value of Goodman-Kruskal index. It is noted that we do not examine Goodman-Kruskal index using biological distances (biological cluster validation), even for few partitions, due to its higher computational complexity as compared with that of using statistical distances (statistical cluster validation), shown before. Also, in Figure 34 the behavior of votes and Rand index as a function of a large range of partition values, concerning Sotiriou's data set is depicted. It is shown that the partition with 110 clusters is a good choice as optimal partition of all candidates. As far as all implemented similarity indices are concerned, for

# Clusters	Votes
50	8.6667
60	7.0000
70	7.6667
100	7.3333
110	7.0000

Table 25 : Sotiriou's data set statistical cluster validation concerning statistical clustering.

# Clusters	Votes
50	9.6667
70	8.0000
80	8.3333
90	8.0000
100	7.3333

Table 26 : Sotiriou's data set BP biological cluster validation concerning statistical clustering.

# Clusters	Votes
50	10.3333
60	10.6667
80	7.0000

Table 27 : Sotiriou's data set MF biological cluster validation concerning statistical clustering.

# Clusters	Votes
50	9.3333
60	8.3333
70	9.0000
100	6.6667

Table 28 : Sotiriou's data set combined BP and MF biological cluster validation concerning statistical clustering.

# Clusters	R
50	0.8986
60	0.9013
70	0.9068
80	0.9089
90	0.9117
100	0.9148
110	0.9159

Table 29 : Sotiriou's data set-based candidate optimal statistical partitions.

this estimated optimal statistical partition compared with the KEGG1 biological partition it holds:

Rand index = 0.9159

Hubert index = 0.8318

Corrected Rand index = 8.0816e-004

So, the optimal statistical partition based on Sotiriou's data set converges strongly to the biological partition obtained from the first method, i.e. from KEGG1 biological clustering. These results confirm our previous choice for the optimal Sotiriou's data set-based statistical partition.

# Clusters	GK Index
20	-1.0253
50	-0.6788
100	0.0486
150	0.1471
200	1.5084

Table 30 : Goodman-Kruskal (GK) index values for some partitions based on statistical cluster validation concerning statistical clustering on Sotiriou's data set.

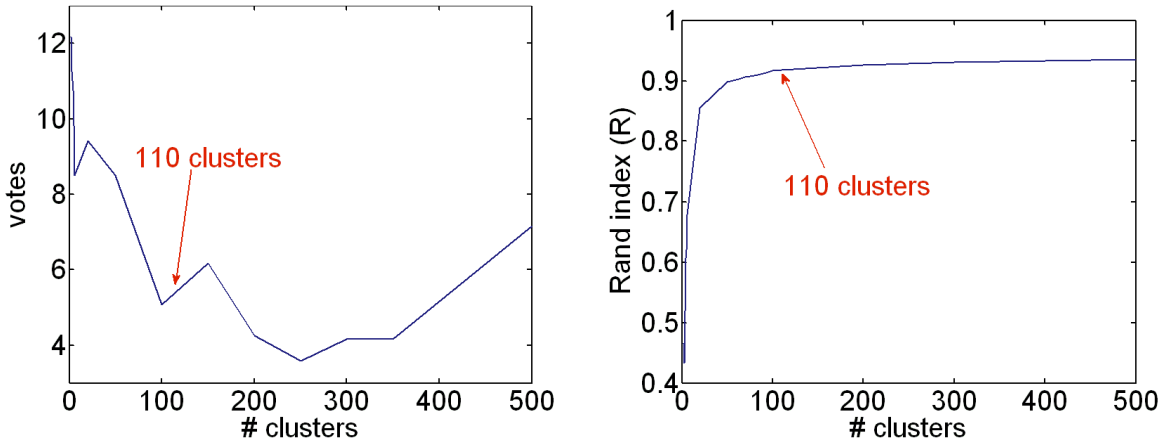


Figure 34 : Behavior of votes and Rand index for different partitions concerning Sotiriou's data set.

4.3.3 Veer's Data Set -Based Statistical Gene Clustering

As far as Veer's data set is concerned:

- From statistical cluster validation concerning statistical clustering, the candidate numbers of statistical clusters (statistical partitions) with the highest votes are shown in Table 31.

- From BP biological cluster validation concerning statistical clustering, the candidate numbers of statistical clusters (statistical partitions) with the highest votes are shown in Table 32.
- From MF biological cluster validation concerning statistical clustering, the candidate numbers of statistical clusters (statistical partitions) with the highest votes are shown in Table 33.
- From combined BP and MF biological cluster validation concerning statistical clustering, the candidate numbers of statistical clusters (statistical partitions) with the highest votes are shown in Table 34.

It is noted that each partition does not contain empty clusters. Next, we compare independently the candidate optimal statistical partitions obtained from Veer's data set (shown in Tables 31-34), with the biological partition obtained from KEGG knowledge through the first method, i.e. through KEGG1 biological clustering. For this purpose we use Rand index. Thus, we estimate the optimal statistical partition that converges most to the biological partition obtained from KEGG knowledge through the first method. The obtained results are summarized in Table 35. From the results in Table 35, we estimate as the optimal Veer's data set-based statistical partition, the one with the maximum Rand index (R), i.e. the partition with 150 clusters. Furthermore, from Table 36 we see that the above partition with 150 clusters is a good choice as optimal partition, since it corresponds to large value of Goodman-Kruskal index. We do not examine Goodman-Kruskal index using biological distances (biological cluster validation), even for few partitions, due to its higher computational complexity as compared with that of using statistical distances (statistical cluster validation) shown before. Also, in Figure 35 the behavior of votes and Rand index as a function of a large range of partition values, concerning Veer's data set is demonstrated, indicating that the partition with 150 clusters is a good choice as optimal partition of all candidates. As far as all implemented similarity indices are concerned, for this optimal statistical partition compared with the KEGG1 biological partition it holds:

Rand index = 0.9186

Hubert index = 0.8371

Corrected Rand index = 0.0065

# Clusters	Votes
50	6.6667
60	10.000
130	6.3333
150	7.6667

Table 31 : Veer's data set statistical cluster validation concerning statistical clustering.

# Clusters	Votes
50	7.6667
60	9.6667
90	7.3333
100	8.6667

Table 32 : Veer's data set BP biological cluster validation concerning statistical clustering.

# Clusters	Votes
50	11.0000
80	8.3333
90	7.0000
110	7.0000

Table 33 : Veer's data set MF biological cluster validation concerning statistical clustering.

# Clusters	Votes
50	11.0000
60	8.3333
80	7.6667
110	6.3333

Table 34 : Veer's data set combined BP and MF biological cluster validation concerning statistical clustering.

# Clusters	R
50	0.8874
60	0.9010
80	0.9068
90	0.9096
100	0.9154
110	0.9163
130	0.9173
150	0.9186

Table 35 : Veer's data set-based candidate optimal statistical partitions.

So, the optimal Veer's data set-based statistical partition converges strongly to the biological partition obtained from the first method, i.e. from KEGG1 biological clustering. These results validate our choice for the optimal Veer's data set-based statistical partition, made before.

# Clusters	GK Index
20	-1.2710
50	-0.0399
100	-0.6355
150	1.0893
200	0.8571

Table 36 : Goodman-Kruskal (GK) index values for some partitions based on statistical cluster validation concerning statistical clustering on Veer's data set.

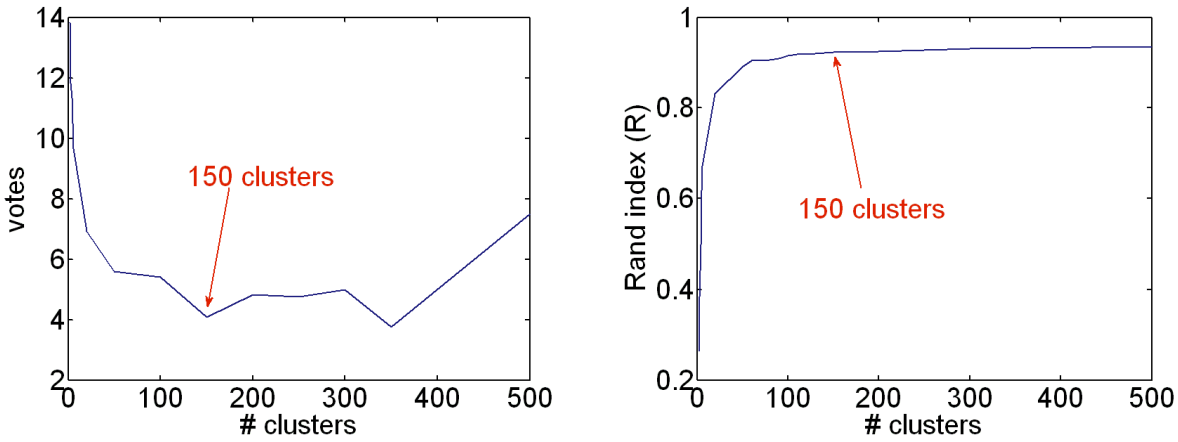


Figure 35 : Behavior of votes and Rand index for different partitions concerning Veer's data set.

Overall, we conclude that the optimal statistical partitions, obtained regardless from each of the three available data sets, converge strongly to the biological partition obtained from the first method (KEGG1 biological clustering). That means that data sets –based statistical clustering leads to gene clusters which are biologically meaningful.

From all the above figures we see that there is a variety of candidate optimal statistical partitions with the highest votes. However, we select as optimal the partition which has a number of clusters close to that one of the obtained KEGG1 biological partition. We also justify our choice with the behavior of Rand index, as shown in the above figures, where we observe that Rand index has achieved its highest value for this estimated optimal statistical partition. That means that the above choice seems to be the optimal that converges mostly to KEGG1 biological partition. It

is also important to note that as far as votes are concerned, the optimal number of clusters with the highest vote seems to be much smaller than the selected one. The reason might be that after the necessary pre-processing (see Section 3.2), the genes that remain to be clustered might belong to much fewer clusters than all the genes in the available data sets before the pre-processing. Furthermore, we notice that the used similarity indices (i.e. Rand index, Hubert index, corrected Rand index) have a major drawback for our data sets. As we have discussed in Section 3.11, they focus on the correspondence on how gene pairs are classified between two partitions. In our data sets after applying clustering methodology, the gene pairs that belong to the same clusters are much less than those that belong to different clusters. As a result, the more clusters we have the larger similarity indices' value we achieve, since more gene pairs that belong to different clusters appear.

4.3.4 Comparison of Obtained Statistical Partitions

Afterwards, we compare the estimated optimal statistical partitions obtained from the available three data sets among them. The results obtained are shown below.

- Veer's data set-based optimal statistical partition compared with Sorlie's data set-based optimal statistical partition

  Rand index = 0.9553
  Hubert index = 0.9106
  Corrected Rand index = 0.0163

- Veer's data set-based optimal statistical partition compared with Sotiriou's data set-based optimal statistical partition

  Rand index = 0.9530
  Hubert index = 0.9060
  Corrected Rand index = 0.0089

- Sotiriou's data set-based optimal statistical partition compared with Sorlie's data set-based optimal statistical partition

  Rand index = 0.9556
  Hubert index = 0.9112

Corrected Rand index = 0.0108

It is also important to note that, as it has been already mentioned, the Rand (R) and Hubert (H) similarity indices are equivalent after correction for agreement due to chance. From the above results, we conclude that the optimal statistical partitions, obtained regardless from each of the three available data sets, are very similar. That means that regardless the data set, the optimal statistical partition of certain genes remains almost the same.

4.4 GO–Based Biological Gene Clustering

Another approach of executing hard c-means clustering is based on the calculated biological distances, using one GO hierarchy each time (*GO biological clustering*). Here we do not use none of the available data sets with genes' expression values. We take advantage only of the calculated biological distances based on ontologies organization. The genes of our interest are the 946 genes that are obtained after the necessary pre-processing (see Section 3.2). To validate the biological partition emerged from GO biological clustering approach, a GO biological cluster validation is required, as discussed previously. It is noted that a normalization and a weighted voting strategy are also applied to improve the GO biological cluster validation. As the red arrows in Figure 28 imply, the clustering method and the validity measures are executed multiple times with different input parameter values, i.e. number of clusters, until some candidate optimal biological partitions are obtained. Since a large number of clusters and very small clusters should be avoided, we focus on a specific area of candidate optimal numbers of biological clusters, i.e. candidate optimal biological partitions, and we finally select the candidates with the highest votes. Then, Rand index (see Section 3.11), which is a similarity measure between GO candidate biological partitions and KEGG1 biological partition, is used to estimate the optimal GO biological partitions.

It is also important to note, as it has been also mentioned in Section 3.3, that due to Wu and Palmer method's high computational complexity, we keep for each gene only few BP, MF or combined BP and MF terms to calculate the BP, MF or combined BP and MF biological distances respectively. This may result in a biased results interpretation. In this section we do not examine Goodman-Kruskal index using biological distances (biological cluster validation), even

for few partitions, due to its higher computational complexity as compared with that of using statistical distances (statistical cluster validation), shown in Section 4.3.

4.4.1 BP-Based Biological Gene Clustering

From BP biological cluster validation concerning BP biological clustering, the candidate numbers of biological clusters (biological partitions) with the highest votes are shown in Table 37. It is noted that in each partition there are not empty clusters. Next, we compare the biological partitions obtained from BP hierarchy with the biological partition obtained from KEGG knowledge via the first method. Thus, we estimate the optimal BP hierarchy-based biological partition that converges most to the biological partition obtained from KEGG knowledge via the first method. The results obtained are shown in Table 38. From the results in Table 38, we estimate as the optimal BP hierarchy-based biological partition, the one with the maximum Rand index (R), i.e. the partition with 100 clusters. Furthermore, in Figure 36 the behavior of votes and Rand index as a function of a large range of partition values, concerning BP hierarchy, indicates that the partition with 100 clusters is a good choice as optimal partition of all candidates. As far as all implemented similarity indices are concerned, for this estimated optimal biological partition compared with the KEGG1 biological partition it holds:

Rand index = 0.7087

Hubert index = 0.4173

Corrected Rand index = -0.0156

Hence, the optimal BP hierarchy-based biological partition is not similar enough to the biological partition obtained from the first method, i.e. from KEGG1 biological clustering. Thus, we conclude that using BP hierarchy is not a good alternative way to obtain biological clusters.

# Clusters	**Votes**
50	8.6667
70	9.0000
80	7.0000
100	7.3333

Table 37 : BP biological cluster validation concerning BP biological clustering.

# Clusters	R
50	0.6570
70	0.6966
80	0.6973
100	0.7087

Table 38 : BP hierarchy-based candidate optimal biological partitions.

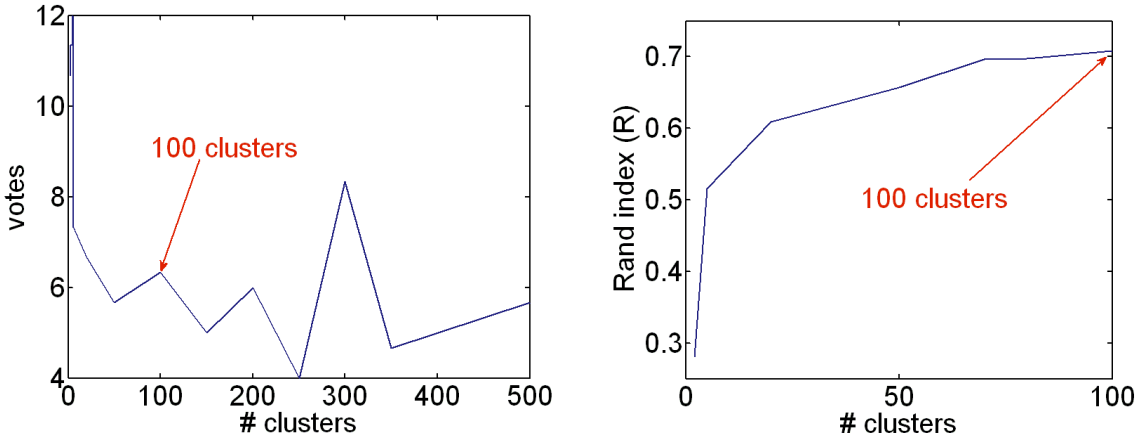


Figure 36 : Behavior of votes and Rand index for different partitions concerning BP hierarchy.

4.4.2 MF-Based Biological Gene Clustering

From MF biological cluster validation concerning MF biological clustering, the candidate number of biological clusters (biological partitions) with the highest votes are shown in Table 39. It is noted that each partition does not contains empty clusters. Next, we compare the biological partitions obtained from MF hierarchy with the biological partition obtained from KEGG knowledge via the first method. Thus, we estimate the optimal MF hierarchy-based biological partition that converges most to the biological partition obtained from KEGG knowledge via the first method. The results are shown in Table 40. From the results in Table 40, we estimate as the optimal MF hierarchy-based biological partition, the one with the maximum Rand index (R), i.e. the partition with 130 clusters. Furthermore, in Figure 37 the behavior of votes and Rand index as a function of a large range of partition values, concerning MF hierarchy, indicates that the partition with 130 clusters is a good choice as optimal partition of all candidates. As far as all im-

plemented similarity indices are concerned, for this estimated optimal biological partition compared with the KEGG1 biological partition it holds:

Rand index = 0.5803

Hubert index = 0.1605

Corrected Rand index = -0.0047

Hence, the optimal MF-based biological partition is not similar at all to the biological partition obtained from the first method, i.e. from KEGG1 biological clustering. Thus, we conclude that using MF hierarchy is not a good alternative way to obtain biological clusters.

# Clusters	Votes
50	6.6667
60	6.6667
70	8.0000
80	8.3333
110	8.0000
130	6.0000

Table 39 : MF biological cluster validation concerning MF biological clustering.

# Clusters	R
70	0.4957
80	0.5305
110	0.5536
130	0.5803

Table 40 : MF hierarchy-based candidate optimal biological partitions.

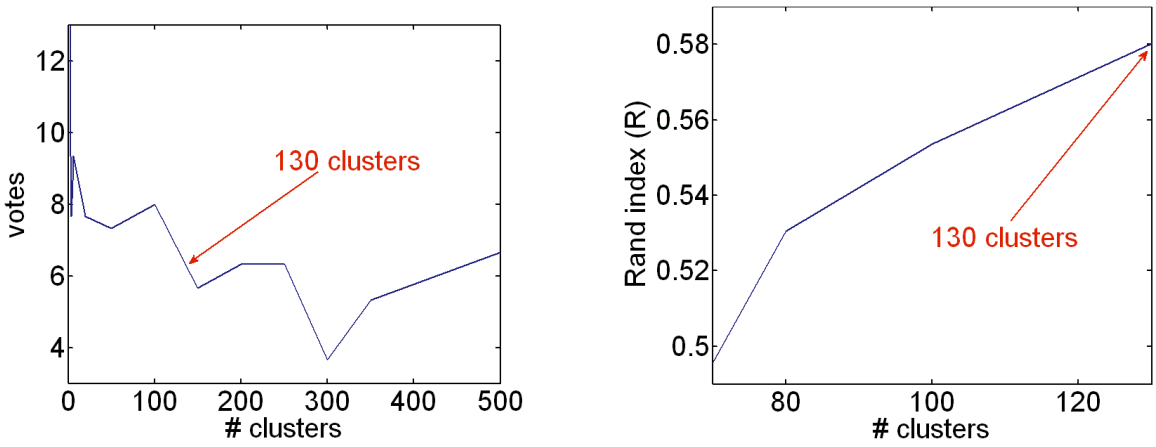


Figure 37 : Behavior of votes and Rand index for different partitions concerning MF hierarchy.

4.4.3 Combined BP and MF -Based Biological Gene Clustering

From the combined BP and MF biological cluster validation concerning combined BP and MF biological clustering, the candidate numbers of biological clusters (biological partitions) with the highest votes are shown in Table 41. It is noted that each partition does not contain empty clusters. Next, we compare the biological partitions obtained from combined BP and MF hierarchy with the biological partition obtained from KEGG knowledge via the first method. Thus, we estimate the optimal combined BP and MF-based biological partition that converges most to the biological partition obtained from KEGG knowledge via the first method. The results obtained are shown in Table 42. From the results in Table 42, we estimate as the optimal combined BP and MF hierarchy -based biological partition, the one with the maximum Rand index (R), i.e. the partition with 100 clusters. Furthermore, in Figure 38 the behavior of votes and Rand index as a function of a large range of partition values, concerning combined BP and MF hierarchy, indicating that the partition with 100 clusters is a good choice as optimal partition of all candidates. As far as all implemented similarity indices are concerned, for this estimated optimal biological partition compared with the KEGG1 biological partition it holds:

Rand index = 0.8704

Hubert index = 0.7407

Corrected Rand index = 0.0282

Hence, the optimal combined BP and MF hierarchy-based biological partition converges enough to the biological partition obtained from the first method, i.e. from KEGG1 biological clustering.

These results validate our previous choice for the optimal combined BP and MF -based biological partition. Also, we conclude that using combined BP and MF hierarchy is a good alternative way to obtain biological clusters. Furthermore, Figure 39 presents all the obtained results of Rand index. As we can see $1+4<1+3<1+3+4$, which implies that better results can be obtained using the combined BP and MF hierarchy, rather than using the other two hierarchies.

# Clusters	Votes
50	6.3333
60	8.0000
70	8.3333
80	6.6667
100	6.6667

Table 41 : Combined BP and MF biological cluster validation concerning combined BP and MF biological clustering.

# Clusters	R
60	0.8658
70	0.8593
80	0.8597
100	0.8704

Table 42 : Combined BP and MF hierarchy-based candidate optimal biological partitions.

From all the above figures we see that there is a variety of candidate optimal GO-based biological partitions with the highest votes. However, we select as optimal the biological partition which has a number of clusters close to that one of the obtained KEGG1 biological partition. It is also important to note that as far as votes are concerned, the optimal number of clusters with the highest vote seems to be much smaller than the selected one. The reason might be that after the necessary pre-processing (see Section 3.2), the genes that remain to be clustered might belong to much fewer clusters than all the genes in the available data sets before the pre-processing.

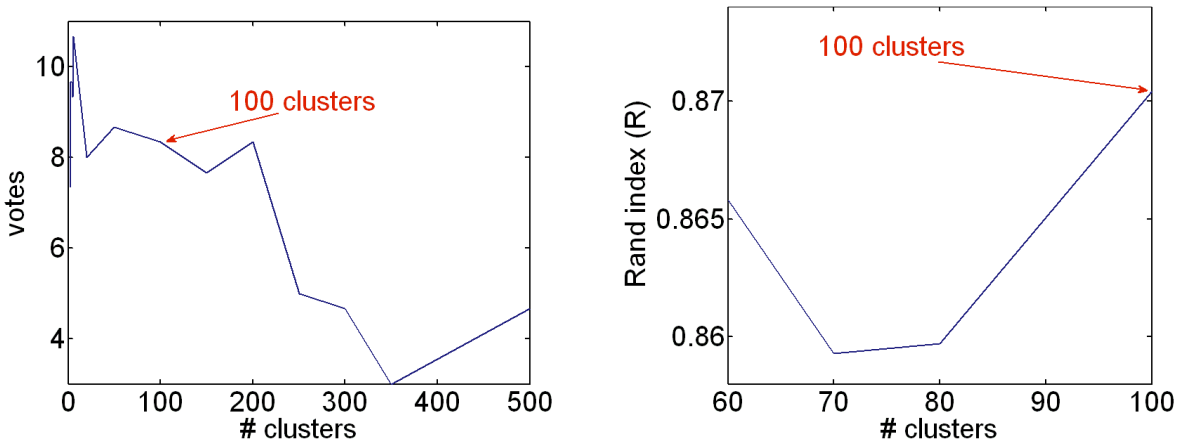


Figure 38 : Behavior of votes and Rand index for different partitions concerning combined BP and MF hierarchy.

4.4.4 Comparison of Obtained Biological Partitions

Next, we compare the optimal biological partitions obtained from the available GO knowledge, using one GO hierarchy each time, among them. The results obtained are shown below.

- BP-based optimal biological partition compared with MF-based optimal biological partition

  Rand index = 0.5422

  Hubert index = 0.0843

  Corrected Rand index = 0.0096

- BP-based optimal biological partition compared with combined BP and MF -based optimal biological partition

  Rand index = 0.7477

  Hubert index = 0.4954

  Corrected Rand index = 0.1042

  It is noted that BP hierarchy is a subset of combined BP and MF hierarchy. Thus, we expect that Rand index < 1 and Hubert index < 1. We can also support this from Figure 39, since $1+2+3<1+2+3+4$.

- MF-based optimal biological partition compared with combined BP and MF -based optimal biological partition

  Rand index = 0.5786

Hubert index = 0.1571

Corrected Rand index = 0.0371

It is noted that MF hierarchy is a subset of combined BP and MF hierarchy. Thus, we expect that Rand index < 1 and Hubert index < 1. We can also crosscheck this from Figure 39, since $1+2+4<1+2+3+4$.

From the above results, we can conclude that the obtained optimal GO-based biological partitions do not resemble at all each other. Figure 39 also presents all the obtained results of Rand index. It is noted that $1+2<1+2+4<1+2+3$, according to our results.

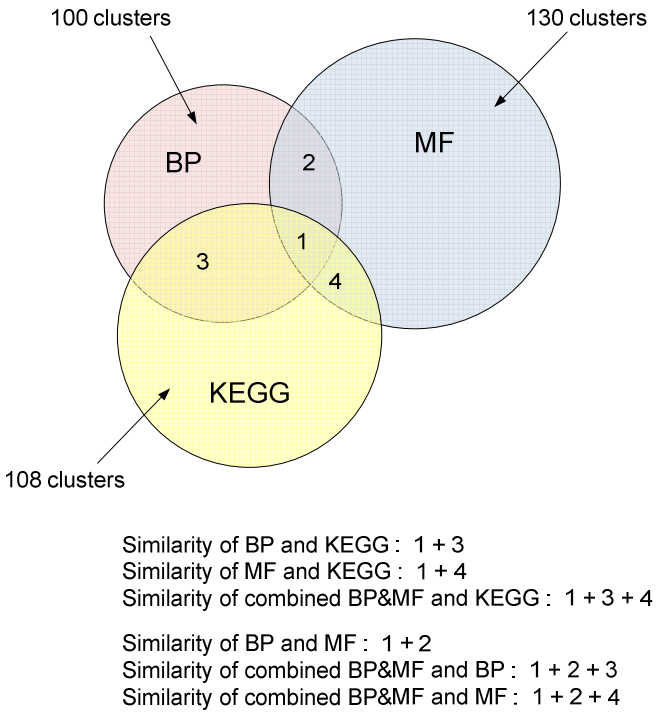


Figure 39 : An illustrative example of all obtained results of Rand index.

4.5 Comparison between Gene Clustering Based on the GO and Based on Statistical Knowledge

Finally, we compare the optimal statistical partitions obtained from the three data sets, presented in Section 4.3, with the optimal biological partitions obtained from the GO, i.e. the three GO hierarchies, presented in Section 4.4. The results obtained are shown below.

- Sorlie's data set-based optimal statistical partition compared with BP-based optimal biological partition
  Rand index = 0.7314
  Hubert index = 0.4628
  Corrected Rand index = -0.0020

- Veer's data set-based optimal statistical partition compared with BP-based optimal biological partition
  Rand index = 0.7347
  Hubert index = 0.4694
  Corrected Rand index = -0.0048

- Sotiriou's data set-based optimal statistical partition compared with BP-based optimal biological partition
  Rand index = 0.7142
  Hubert index = 0.4285
  Corrected Rand index = 0.0027

- Sorlie's data set-based optimal statistical partition compared with MF-based optimal biological partition
  Rand index = 0.5835
  Hubert index = 0.1671
  Corrected Rand index = 0.0055

- Veer's data set-based optimal statistical partition compared with MF-based optimal biological partition
  Rand index = 0.5802
  Hubert index = 0.1605
  Corrected Rand index = 7.1032e-004

- Sotiriou's data set-based optimal statistical partition compared with MF-based optimal biological partition
  Rand index = 0.5806

Hubert index = 0.1612
Corrected Rand index = 0.0031

- Sorlie's data set-based optimal statistical partition compared with combined BP and MF -based optimal biological partition
  Rand index = 0.8941
  Hubert index = 0.7882
  Corrected Rand index = 0.0012

- Veer's data set-based optimal statistical partition compared with combined BP and MF -based optimal biological partition
  Rand index = 0.9030
  Hubert index = 0.8060
  Corrected Rand index = -7.8115e-004

- Sotiriou's data set-based optimal statistical partition compared with combined BP and MF -based optimal biological partition
  Rand index = 0.8969
  Hubert index = 0.7939
  Corrected Rand index = 0.0019

From the above results, we conclude that the obtained optimal GO-based biological partition using the combined BP and MF hierarchy is very similar with the optimal statistical partitions obtained from the three data sets. However, using the BP or MF hierarchy doesn't lead to desired results. Thus, we confirm again that using only combined BP and MF hierarchy is a good alternative way to obtain a gene partition.

4.6 Summary

In this chapter, we present the obtained results from the gene clustering methodologies implemented in this thesis (see Figure 28). We also compare and interpret the results and we finally make meaningful biological conclusions. We conclude that there are four different clustering approaches to obtain biological partitions from the available KEGG PWs knowledge. Also, statis-

tical partitions based on data sets converge strongly to the biological partition obtained from the first clustering method based on KEGG PWs knowledge. These statistical partitions obtained from the data sets are very similar each other. Furthermore, one major conclusion is that using the GO hierarchies does not lead to biological meaningful partitions, except using the combined BP and MF hierarchy, which takes into consideration both BP and MF hierarchy knowledge. Finally, these biological partitions based on the GO hierarchies don’t resemble each other.

CHAPTER 5: DISCUSSION AND OPEN PROBLEMS

5.1 Introduction

In this chapter, we summarize the main meaningful biological conclusions obtained in Chapter 4. We also introduce some ideas for further research. Furthermore, we suggest some guidelines about the implemented gene clustering methodologies, which might lead to better results and then, to more meaningful biological conclusions.

5.2 Main Conclusions

In this section we summarize in brief the findings in Chapter 4. It has been illustrated that it is feasible to obtain biological partitions from the available KEGG PWs knowledge via four different methods. In particular, the obtained optimal biological partitions based on KEGG PWs knowledge are the following:

- the biological partition with 108 clusters obtained from the KEGG1 method
- the biological partition with 130 clusters obtained from the KEGG2 biological clustering method using Correlation distance metric
- the biological partition with 80 clusters obtained from the KEGG2 biological clustering method using Euclidean distance metric
- the biological partition with 100 clusters obtained from the KEGG3 biological clustering method

It has been also shown that using the preceding methods, genes are eventually characterized as “well-clustered”.

It is also observed that the optimal statistical partitions, obtained independently from each of the three available data sets, converge strongly to the biological partition obtained from the first method (KEGG1 biological clustering). This reveals that data sets –based statistical clustering leads to gene clusters which are biologically meaningful. In particular, the obtained optimal statistical partitions are:

- the statistical partition with 90 clusters based on Sorlie's data set
- the statistical partition with 110 clusters based on Sotiriou's data set
- the statistical partition with 150 clusters based on Veer's data set

Furthermore, the optimal statistical partitions obtained independently from each of the three available data sets are very similar each other, implying that regardless the data set, the optimal statistical partition of certain genes remains almost the same. However, as we have discussed in the previous Chapter in Section 4.3.3, it is important to mention that the used validity measures focus on the correspondence on how gene pairs are classified between two partitions. In our data sets after applying clustering methodology, the gene pairs that belong to the same clusters are much less than those that belong to different clusters. As a result, the more clusters we have the larger similarity indices' value we achieve, since more gene pairs that belong to different clusters appear. The above similarity indices' drawback might confuse us to choose the optimal statistical partition.

As far as using GO hierarchies, the obtained optimal biological partitions based on GO hierarchies knowledge are the following:

- the biological partition with 100 clusters based on the BP hierarchy
- the biological partition with 130 clusters based on the MF hierarchy
- the biological partition with 100 clusters based on the combined BP and MF hierarchy

We have shown that better results can be obtained using the combined BP and MF hierarchy, rather than using the BP or MF hierarchy. It has been also observed that the obtained optimal GO-based biological partitions do not resemble at all each other. Finally, we conclude that the obtained optimal GO-based biological partition using the combined BP and MF hierarchy is very similar with the obtained optimal statistical partitions from the three data sets. However, using the BP or MF hierarchy doesn't lead to such desired results. Thus, it is demonstrated again that using only combined BP and MF hierarchy a reliable gene partition can be obtained.

In addition, it is important to note that as far as the applied weighed voting strategy is concerned, the optimal number of statistical or GO-based biological clusters with the highest vote seems to be much smaller than the selected one. The reason might be that after the necessary pre-processing (see Section 3.2), the genes that remain to be clustered might belong to much fewer clusters than all the genes in the available data sets before the pre-processing.

We finally conclude that it is feasible to obtain gene partition with biological significance via the available data sets (i.e. Sorlie's, Sotiriou's and Veer's data set) with genes' expression values. On the other hand, using available GO knowledge doesn't lead to meaningful conclusions. Except using the combined BP and MF hierarchy that leads to successful gene clusters, using BP or MF hierarchy doesn't lead to such desired results. In the next section we introduce a variety of ideas that might improve the implemented gene clustering methodologies.

5.3 Further Research

Taking advantage of the implemented gene clustering methodology, future research efforts might focus on supporting statistical clustering of a data set. In particular, it is interesting to examine how statistical clustering of a data set can be influenced by the biologically and statistically relevant clusters of another data set with common genes. This can be done by incorporating the biologically and statistically relevant clusters of a data set, which are obtained by the implemented methodology, in the clustering algorithm of another data set with common genes.

Also, another idea for further research could be to improve the Wu and Palmer's method, discussed in Section 3.3. In current study, only a maximum number of five terms per gene were used to calculate the gene distances. It would be interesting to optimize this algorithm and include more terms in order to calculate more accurate biological distances. This will provide more reliable biological conclusions. It will be also interesting to select the five most biologically meaningful terms per gene to calculate the gene distances. This might lead to more accurate biological distances too.

Furthermore, it is common that the protein products of genes are involved in multiple biological processes and thus the gene producing these proteins can be co-regulated in different ways under different conditions. When a gene experiences differential co-regulation in different sam-

ples of the same dataset as a result of being involved in differing functional relationships, traditional clustering approaches are not flexible to represent this behavior. Hence, fuzzy c-means, presented in Section 3.5, would be a suggested approach of gene clustering method, since it is capable to assign genes to multiple clusters, which is a more appropriate representation of the behavior of genes.

In addition as we have said in the previous Chapter, as far as validity assessment system is concerned, the optimal number of statistical or GO-based biological clusters with the highest vote seems to be much smaller than the selected one. One idea is to examine if these few clusters, which are shown as optimal, have biological significance and then, to make some important conclusions about this case.

Another idea would be to implement some other similarity measures with different basic idea than those already applied, in order to justify or not our choice of the optimal statistical or biological partition (KEGG or GO –based biological partition) each time. Thus, we will examine the influence of the aforementioned drawback that the used similarity measures have.

Finally, to aid the interpretation of GO, a set of general GO terms called GOSlim terms (see Section 2.4) is defined for various organisms as well as generic use. The use of GOSlim terms can be seen as a way to determine the similarity of genes. Thus, considering more terms, since GO slims are cut-down versions of the GO ontologies containing a subset of the terms in the whole GO, the performance of Wu and Palmer's method can be improved.

5.4 Summary

In this chapter, we summarize in brief the conclusions in Chapter 4. Furthermore, we introduce some novel ideas to motivate further research on how statistical clustering of data sets can be influenced by the biologically and statistically relevant clusters of another data set with common genes, which is obtained by the implemented methodology. We also suggest some guidelines about the implemented gene clustering methodologies, which might lead to better results and then, to more meaningful biological conclusions. Implementing fuzzy c-means instead of hard c-means, taking advantage of GOSlim terms or giving more GO terms as input parameter in Wu and Palmer's method, are some of the proposed guidelines.

REFERENCES

[1] E. Backer and A. K. Jain, "A Clustering Performance Measure Based on Fuzzy Set Decomposition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-3, no. 1, pp. 66-75, Jan. 1981

[2] B. Everitt, *Cluster Analysis*. Halsted Press, New York, 1993

[3] L. Kaufman and P. J. Rousseeuw, *Finding Groups in Data: An Introduction to Cluster Analysis*. Wiley, 1990

[4] A. K. Jain, M. N. Murty and P. J. Flynn, "Data Clustering: A Review," *ACM Computing Surveys*, vol. 31, no. 3, pp. 264-323, September 1999

[5] The Gene Ontology (www.geneontology.org)

[6] Wikipedia, the Free Encyclopedia (http://en.wikipedia.org/wiki/)

[7] J. Fitch and B. Sokhansanj, "Genomic Engineering: Moving beyond DNA. Sequence to Function," *Proceedings of the IEEE* , vol. 88, no. 12, pp. 1949-1971, Dec. 2000

[8] N. Bolshakova and F. Azuaje, "Cluster Validation Techniques for Genome Expression Data," *Signal Processing*, vol. 83, no. 4, pp. 825–833, 2003

[9] N. Bolshakova and F. Azuaje, "Machaon CVE: Cluster Validation for Gene Expression Data," *Bioinformatics*, vol. 19, no. 18, pp. 2494–2495, 2003

[10] KEGG: Kyoto Encyclopedia of Genes and Genomes (www.genome.jp/kegg/)

[11] R. Xu and D. C. Wunsch II, *Clustering*. IEEE Press Series on Computational Intelligence, Wiley, New York, 2009

[12] Jose Valente de Oliveira and W. Pedrycz, *Advances in Fuzzy Clustering and its Applications*. Wiley, New York, 2007

[13] N. Bolshakova and F. Azuaje, "Estimating the Number of Clusters in DNA Microarray Data," *Methods Inf. Med.*, vol. 45, no. 2, pp. 153-157, 2006

[14] L. Hubert and J. Schultz, "Quadratic Assignment as a General Data Analysis Strategy," *British Journal of Mathematical and Statistical Psychology*, vol. 29, pp. 190-241, 1976

[15] L. Goodman and W. Kruskal, "Measures of Associations for Cross-Validations," *Journal of American Statistical Association*, vol. 49 , pp. 732–764, 1954

[16] N. Bolshakova and F. Azuaje, "Improving Expression Data Mining through Cluster Validation," *4th International IEEE EMBS Special Topic Conference on Information Technology Applications in Biomedicine*, pp. 19- 22, 24-26, April 2003

[17] P. J. Rousseeuw, "Silhouettes: a Graphical Aid to the Interpretation and Validation of Cluster Analysis," *J. Comp Appl.Math*., vol. 20, 1987

[18] J. C. Dunn, "Well-Separated Clusters and Optimal Fuzzy Partitions," *Journal of Cybernetics*, vol. 4, pp. 95-104, 1974

[19] N. Bolshakova and F. Azuaje, "Cluster Validation Techniques for Genome Expression Data," *Signal Processing*, vol. 83, no. 4, pp. 825-833, 2003

[20] Z. Wu and M. Palmer, "Verb Semantics and Lexical Selection," *32nd Annual Meeting of the Association for Computational Linguistics*, pp. 133 -138, New Mexico, 1994

[21] N. Bolshakova, F. Azuaje and P. Cunningham, "Incorporating Biological Domain Knowledge into Cluster Validity Assessment," *4th European Workshop on Evolutionary Computation and Machine Learning in Bioinformatics*, vol. 3907, pp. 13-22, 2006

[22] P. Resnik, "Using Information Content to Evaluate Semantic Similarity in a Taxonomy," *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI)*, vol. 1, pp. 448-453, 1995

[23] A. Budanitsky and G. Hirst, "Semantic Distance in WordNet: An Experimental, Application-Oriented Evaluation of Five Measures," *Proc. of Workshop on Word Net and Other Lexical Resources*, Pittsburgh, 2001

[24] P. Resnik and M. Diab, "Measuring Verb Similarity," *Proc. of 22nd Annual Meeting of the Cognitive Science Society (COGSCI2000)*, Philadelphia, August 2000

[25] P. Lord, R. Stevens, A. Brass, and C. Goble, “Investigating Semantic Similarity Measures across the Gene Ontology: the Relationship Between Sequence and Annotation,” *Bioinformatics*, vol. 19, pp. 1275-1283, 2003

[26] D. Lin, “An Information-Theoretic Definition of Similarity,” *Proc. of 15th International Conference on Machine Learning*, pp. 296-304, San Francisco, 1998

[27] R. Krishnapuram and J. Keller, “The Possibilistic C-Means Algorithm: Insights and Recommendations,” *IEEE Transactions on Fuzzy Systems*, vol. 4, no. 3, pp. 385-393, 1996

[28] P. Drineas , A. Frieze , R. Kannan , S. Vempala and V. Vinay, “Clustering Large Graphs via the Singular Value Decomposition,” *Machine Learning*, vol. 56, no. 1-3, pp. 9-33, 2004

[29] G. H. Ball and D. J. Hall, “ISODATA, an Iterative Method of Multivariate Data Analysis and Pattern Classification,” *IEEE International Communications Conference Philadelphia*, vol. 2715, USA, June 1966

[30] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*. Wiley, New York, 1973

[31] M. D. McKay, R. J. Beckman and W. J. Conover, “A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code,” *Technometrics*, vol. 21, no. 2, pp. 239–245, 1979

[32] G. H. Ball and D. J. Hall, “A Clustering Technique for Summarizing Multivariate Data,” *Behavioral Science*, vol. 12, pp. 153-155, 1967

[33] L. Kaufman and P. J. Rousseeuw, *Finding Groups in Data*. Wiley, UK, 1990

[34] C. Ordonez, “Clustering Binary Data Streams with K-Means,” *Proc. ACM Data Mining and Knowledge Discovery Workshop*, pp. 12-19, 2003

[35] Z. Huang, “Extensions to the K-Means Algorithm for Clustering Large Data Sets with Categorical Values,” *Data Mining and Knowledge Discovery*, vol. 2, no. 3, pp. 283–304, USA, 1998

[36] S. K. Gupta, K. S. Rao and V. Bhatnagar, "K-Means Clustering Algorithm for Categorical Attributes," *Proceedings of the 1st International Conference on Data Warehousing and Knowledge Discovery*, vol. 1676, pp. 203-208, 1999

[37] D. Charalampidis, "A Modified K-Means Algorithm for Circular Invariant Clustering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 12, pp. 1856-1865, Dec. 2005

[38] G. Peters, "Some Refinements of Rough K-Means Clustering," *Pattern Recognition*, vol. 39, no. 8, p. 1481-1491, August 2006

[39] W.M. Rand, "Objective Criteria for the Evaluation of Clustering Methods," *Journal of the American Statistical Association*, vol. 66, pp. 846-850, 1971

[40] R. L. Brennan and R. J. Light, "Measuring Agreement when two Observers Classify People into Categories not Defined in Advance," *British Journal of Mathematical and Statistical Psychology*, vol. 27, pp. 154–163, September 1974

[41] L. J. Hubert, "Inference Procedures for the Evaluation and Comparison of Proximity Matrices," *In Numerical Taxonomy*, pp. 209-228, New York, 1983

[42] B. G. Mirkin and L. B. Chernyi, "Measurement of the Distance Between Distinct Partitions of a Finite Set of Objects," *Automation and Remote Control*, vol. 31, pp. 786–792, 1970

[43] P. Arabie and S. A. Boorman, "Multidimensional Scaling of Measures of Distance Between Partitions," *Journal of Mathematical Psychology*, vol. 10, pp. 148-203, 1973

[44] L. Hubert and P. Arabie, "Comparing partitions," *Journal of Classification*, vol. 2, no. 1, pp. 193-218, 1985

[45] A. N. Albatineh, M. Niewiadomska-Bugaj and D. Mihalko, "On Similarity Indices and Correction for Chance Agreement," *Journal of Classification*, vol. 23, pp. 301-313, 2006

[46] A. N. Albatineh, "On Similarity Measures for Cluster Analysis," *PhD Dissertation, Kalamazoo, MI: Western Michigan University*, 2004

[47] A. Ben-Dor, L. Bruhn, N. Friedman, I. Nachman, M. Schummer and Z. Yakhini, "Tissue Classification with Gene Expression Profiles," *Journal of Computational Biology*, vol. 7, pp. 559–584, May 2000

[48] National Human Genome Research Institute (www.genome.gov)

[49] A. Jain and R. Dubes, *Algorithms for Clustering Data*. Prentice Hall, 1988

[50] S. Theodoridis and K. Koutroumbas, *Pattern Recognition*. Academic Press, San Diego, 2006

[51] G. Karypis, E. Han and V. Kumar, "Chameleon: Hierarchical Clustering Using Dynamic Modeling," *IEEE Computer*, vol. 32, no. 8, pp. 68–75, 1999

[52] R. Sharan and R. Shamir, "CLICK: A Clustering Algorithm with Applications to Gene Expression Analysis," *In Proceedings of 8th International Conference on Intelligent Systems for Molecular Biology (ISMB)*, AAAI Press, pp. 307–316, 2000

[53] J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*. Plenum Press, New York, 1981

[54] N. Pal, K. Pal, J. Keller and J. Bezdek, "A Possibilistic Fuzzy C-Means Clustering Algorithm," *IEEE Transactions on Fuzzy Systems*, vol. 13, no. 4, pp. 517–530, 2005

[55] R. Yager and D. Filev, "Approximate Clustering via the Mountain Method," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 24, no. 8, pp. 1279-1284, Aug. 1994

[56] R. Michalski and R. E. Stepp, *Machine Learning: An Artificial Intelligence Approach.* Tioga Press, 1983, Chapter 11

[57] G. Biswas, J. B. Weinberg and D. Fisher, "ITERATE: A Conceptual Clustering Algorithm for Data Mining," *IEEE Transactions on Systems, Man, and Cybernetics – Part C: Applications and Reviews*, vol. 28, no. 2, pp. 219–230, 1998

[58] R. E. Neapolitan, *Learning Bayesian Networks*. Prentice Hall, Upper Saddle River, NJ, 2004

[59] A. D. Diehl, J. A. Lee, R. H. Scheuermann and J. A. Blake, "Ontology Development for Biological Systems: Immunology," *Bioinformatics*, vol. 23, no. 7, pp. 913–915, 2007

[60] S. Carbon, A. Ireland, C. J. Mungall, S. Shu, B. Marshall and S. Lewis, "AmiGO: Online Access to Ontology and Annotation Data," *Bioinformatics*, vol. 25, no. 2, pp. 288–289, 2008

[61] J. Day-Richter, M. A. Harris and M. Haendel, "OBO-Edit--An Ontology Editor for Biologists,|" *Bioinformatics*, vol. 23, no. 16, pp. 2189–2200, 2007

[62] The Gene Ontology Consortium, "Gene Ontology: Tool for the Unification of Biology," *Nature Genetics*, vol. 25, no. 1, pp. 25–29, 2000

[63] L. A. Zadeh, "Fuzzy Sets," *Information Control*, vol. 8, pp. 338–353, 1965

[64] J. C. Dunn, "A Fuzzy Relative of the ISODATA Process and its Use in Detecting Compact, Well Separated Clusters," *Journal of Cybernetics*, vol. 3, pp. 95–104, 1974

[65] J. Bezdek, *Fuzzy Mathematics in Pattern Classification*. PhD Thesis Applied Math. Center, Cornell University, Ithaca, USA, 1973

[66] J. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*. Plenum Press, New York, 1981

[67] W. Pedrycz, *Knowledge-Based Clustering: From Data to Information Granules*. Wiley, Holboken, USA, 2005

[68] F. Klawonn, "Understanding the Membership Degrees in Fuzzy Clustering," *In Proc. of the 29th Annual Conference of the German Classification Society, Springer*, pp. 446–454, 2006

[69] H. Timm, C. Borgelt, C. Döring and R. Kruse, "An Extension to Possibilistic Fuzzy Cluster Analysis," *Fuzzy Sets and Systems*, vol. 147, no. 1, pp. 3–16, 2004

[70] F. Höppner, F. Klawonn, R. Kruse and T. Runkler, *Fuzzy Cluster Analysis*. Wiley, Chichester, United Kingdom, 1999

[71] CAMO (www.camo.com/resources/clustering.html)